DOCUMENT RESUME

ED 184 505 IR 008 093

AUTHOR	Caruso, Elaine
TITLE	Training Modules for Use of Scientific and Technical Information Services.
SPONS AGENCY	National Science Foundation, Washington, D.C.
PUB DATE	Dec 77
GRANT	DSI-76-09538
NOTE	141p.; Some appendices and examples will not reproduce clearly.
EDRS PRICE	MF01/PC06 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; Computer Oriented Programs; *Data Bases; *Information Retrieval; Information Services; *Information Systems; Learning Modules; *Online Systems; *Search Strategies; Training
IDENTIFIERS	Scientific and Technical Information

ABSTRACT

Intended for users of online bibliographic retrieval systems, this set of learning modules incorporates instructional aids in three modes: textual, computer assisted learning, and emulations of actual systems (DIALOG and ORBIT) which access a database of segments of over 60 commonly used source files. The textual materials can be used with local or remote access to the target systems, the emulators can be accessed remotely or brought up on any computer system supporting ANSI FORTRAN, and the computer assisted learning and practice (CALP) sequences can be accessed remotely or implemented on a system which maintains a capable lesson designer type language. The modules are followed by sections on database structure and organization, analyses and evaluations of the modules and the collected use data, and the dissemination of the training program. Appendices contain selected pages from two retrieval training manuals, an illustration of use of CALP modules by trainees, excerpts from runs of DIALOG and ORBIT emulators, data collection forms, summaries of system use, abstracts of FORTRAN programs, a description of a proposed cooperative project, and a list of published articles and papers on scientific and technical information services. (FM)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED184505

U.S. DEPARTMENT OF HEALTH.
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

National Science Foundation

Grant No. DSI 76-09538

TRAINING MODULES
FOR USE OF
SCIENTIFIC AND TECHNICAL INFORMATION SERVICES

Report of Work Completed

June 1, 1976 - Nov. 30, 1977

Submitted by

Dorothy Elaine Caruso
Principal Investigator

December 1977

IR008093

Contents

Abstract ii

Competence in Searching iii

Introduction 1

Section 1 Printed Materials 4

Section 2 Computer Assisted Learning and Practice 8

Section 3 Emulation of DIALOG™ and ORBIT™ 15

Exhibits 27

Section 4 Data Base Structure and Organization 53

Section 5 Analysis, Evaluation Display of Data of Use 61

Section 6 Dissemination of the Training Program 71

Table I Report Sections Correlated with Appendices and Authors 2

Figure 1 General System Flow . . . . 12 ff

Figure 2 Augmentation and Extension of Software . . . . 12 ff

Appendices . . . . 73 ff

Abstract

This report summarizes work completed under N.S.F. Grant No. DSI 76-09538, Training Modules for Use of S.T.I. Services, June 1, 1976 - November 30, 1977. It includes a statement of the goals of the project, and of the process followed in the pursuit of those goals; a description of the products of the project; an account of our experience in using the products; and a summary and analysis of results obtained to date in use of the modules in training new users for online bibliographic retrieval systems.

S.T.I. Training Modules for users of online bibliographic retrieval systems incorporate learning aids in three modes: textual, computer assisted learning, and emulations of actual systems (DIALOG™ and ORBIT™, currently) which access a data base of segments of 60+ commonly used source files. The values of the Modules lie in three areas: the careful selection of the capabilities taught, the hospitality of the training materials to the incorporation of new or revised content, and the adaptability of the materials for use in a wide range of environments with or without computer facilities or access to the target systems, and with trainee populations of varying learning styles and motivations. The textual materials can be used as they are, with local or remote access to the emulators or to the target systems; the emulators can be accessed remotely or brought up on any computer system supporting ANSI FORTRAN (currently Digital Equipment Corp. 1175 using 8k of 32-bit word core + FOROTS with 2 million characters of random access mass storage); the computer assisted learning sequences can be accessed remotely or implemented on a system which maintains a capable lesson designer type language.

The economical operation of the emulators of the systems permits extensive practice of the more mechanical skills and ample time for user experimentation and exploration in system capabilities for expression and satisfaction of conceptualized information needs. Basic functions, dictated by data base structures, are emphasized to assure the long term value of the training to the user.

Competence in Searching

User competence will be determined by individual motivation and intelligence; and given these, his knowledge of subject, system, and data base. This knowledge is a function of time available for application at the task, and of the availability of appropriate learning resources; it consists of both understandings and skills.

Our goal in training the end-user is to achieve a "first level of competence":*

1. The user knows and can use the "most used" capabilities of a search program to select files, enter and combine terms, review his progress, and display results.*
2. The user can use the vocabulary negotiation capabilities of the search to retrieve document references of value to him; he can increase or decrease the size of the retrieved document set in a meaningful way by using the logical operators AND, NOT, OR.
3. The user knows the data base(s) most useful to him; he can use documentation provided to determine (a) which data elements of the record can be directly searched; and (b) whether controlled or uncontrolled indexing is used in the fields he searches, and can use available vocabulary aids for choosing useful search terms both online and offline.

---

*See Appendix A, "Hands On Online in a Multi-System, Multi-Data Base Trainer," a manual, for specific commands and for defined second and third levels of competence.

Introduction

The work reported herein has, as its overall goal, the design, development, and evaluation of training materials which would enable users of scientific and technical information to identify appropriate machine-readable data bases and to access those files effectively, using currently available search systems.

Products of the research include (1) printed materials: a training manual and a series of "for practice" question types designed to exemplify particular search system characteristics and data base attributes; (2) a series of online training and practice modules which are coordinated with the printed materials; (3) emulations of two commercially available search services (DIALOG^{TM} and ORBIT^{TM});* and (4) a data base of over 60 machine-readable files, each file approximating in size one issue of each file title included.**

Evaluation of the products of the research has proceeded concurrently with the developmental work; as each version of printed material or module of the computer assisted learning and practice sequence was completed, it was made known to the community of users of the University's computer system, and made freely available for their use, either as individuals or in informal or formal classes conducted by project members or other faculty of the Graduate School of Library and Information Sciences. Data generated by this use was collected automatically by the programs, and by solicitation of freely expressed user reactions and reports of difficulties within the program itself, and in presentations of the program to the various potential user populations.

---

*DIALOG^{TM} is the trademark for the online bibliographic retrieval service offered by Lockheed Retrieval Services, Palo Alto, CA, and ORBIT^{TM} is the name of the service offered by System Development Corp., Santa Monica, CA.

**See Appendix A for a listing of the titles of the files available to the training program. Files were acquired by the CBIS (Campus Based Information Systems)project (partically supported by an N.S.F. Grant) of the Office of Communications Programs of the University of Pittsburgh, Allen Kent, Director.

More objective evaluation and wider dissemination and use of the training program were not attempted as the program was not then complete; such assessment and distribution are the focus of the projected year's work.

This report is organized under the topics suggested in this introduction, and exhibits appended parallel that organization; members of the project group who, while participating to some degree in all aspects of the work, took primary responsibility for particular areas, are identified for each section: (see also Table of Contents for page references)

Table I Report Sections Correlated with Appendices and Authors

Topic	Appendix	Author/Developer
1. Printed Training Materials	A. Training Manual B. Suggestions for Independent Practice	Caruso
2. Computer Assisted Learning and Practice Modules	C. Illustration of Runs of CALP Modules by Trainees	Rathburn
	D. Program Listings of CALP Modules	
3. Emulation of DIALOG and ORBIT Retrieval Systems	E. Illustration of Runs of Emulators by Trainees	Griffiths
	F. Program Listings for Emulations	

Topic	Appendix	Author/Developer
4. Data Base Structure and Organization	G. Program Listings for File Generation and Maintenance	Williams
5. Analysis of Use, and Evaluation of Training Modules	H. Sample of User Attribute Data Collection	Nicholas
Reduction and Display of Automatically Collected Use Data	J. Summary of User History (Preliminary Format)	Scollon
	K. Summary of CALP History (Preliminary Format)	
	L. Abstracts of Fortran Programs	
6. Dissemination of Training Program	M. Online Training Institute,Brochure	Caruso
	N. Description of Proposed Cooperative Project, with Prof. A. Neelameghan, Director of Documentation Training and Research Center	
	O. List of Published Articles and Papers Generated by this Project	

Coordination of these separate elements required continuous interaction between the individuals, since their work was so interdependent: the printed text to coordinate especially closely with the computer assisted learning and practice (CALP) modules; the emulators to implement the commands and capabilities taught in the CALP; and the data base content and structure to support the capabilities built into the emulators. This resulted in the generation of documentations which could be used as reference data by the project members as they worked. Ultimately a detailed complete comparison of the two systems was produced, with capabilities to be taught and implemented on the two emulators (see Appendix A).

A close analysis of commands and capabilities of the systems was necessary to ensure that the emulators were as equivalent in power as possible; and to permit both emulators to share the same service routines (see Section 3). A design goal of the project was that it be easily transportable to other computing environments; use of FORTRAN was thus the chosen language for emulator and data base generation programs (see Sections 3 and 4). Transportability of the Computer Assisted Learning and Practice Modules is currently easily achievable to host systems which support lesson designer languages (see Section 2); however planned encoding of these modules for the DEC PDP 11/40 minisystem will generate more widely usable coding.

The overall design of the CALP and emulator modules is hospitable to the extension of system training and to introduction of additional search systems (see Figure 1, 2 ff. p. 12, and Section 3, p. 24).

Section 1 S.T.I. Training Modules Report

December 1977

Printed Materials
Individual Primarily Responsible: Elaine Caruso*

Related Appendices:

A. "Hands On Online in a Multi-System, Multi-Data Base Trainer"
a manual for users of S.T.I. Training Modules

B. "Suggestions for Independent Practice"
for users of S.T.I. Training Modules

---

*The very valuable assistance of Jean Guasco in critical reading of the printed materials and in coordinating terminology between the printed materials and the computer generated text is gratefully acknowledged

The training program manual, "Hands on Online in a Multi-System, Multi-Data Base Trainer," speaks for itself: its purpose, content, organization, and use. It does so because, as an aid to *independent, individually pursued* study and skill development, these attributes must be made explicit; no human teacher is assumed to be available to interpret and direct the use of the manual itself, or the suggested independent exercises or the tutorial and emulator modules.

The first version of the manual developed concurrently with the computer modules and the process was a continuous one of juggling long textual messages between the printed medium and the computer medium. The two conflicting principles were these: first that the computer should not be used to do what a printed text could do as well (or better!), and second, that all needed information should be available to the trainee, at the time of need, at the training terminal. A series of compromises developed; the rationale for compromise was two-phased:

(1) when long segments of printed explanation were unavoidable, as in the introduction of various commands and rules of statement construction, the text was included both in the manual and in the computer sequence. The manual and the computer manual each referred the trainee to the duplicate provision of the text; at the training terminal the trainee was given the opportunity to skip the textual printout if he chose, either by reason of having the manual beside him, or by reason of previous reading of the text.

(2) when extensive descriptive material on files was needed, the user was frankly referred to printed supportive literature, since this mirrors the "real world" use of the search service.

Extensive descriptions of file content, and directly searchable fields, etc. are provided offline in printed versions for each file of the data base.

The present version of the manual is not a revision of the first version, but completely rewritten, in an attempt to capture in a consistent style all that we had learned in the development and use of the first version.

Our experience with independent trainees exposed a gap between the computer assisted learning and practice modules (see Section 2 of this report) which introduced and gave directed practice in search aspects, and the emulator modules (see Section 3), which allowed the trainee to test his mastery of the search aspect learned. The problem that was revealed was in the nature of the two kinds of processes which go on in using any information system:

1. the conceptualization of an information need, in the terms of the information resource approached for satisfaction of that need (e.g., stating a need for information on effects of diet on child behavior is quite different if the resource is a medical bibliography,or a psychology index, or educational tool, etc.)
2. manipulation of the mechanisms of the information resource (e.g., how to get to the reference desk, catalog, index, or to call up a data base file, browse its index...) to satisfy the properly stated information need.

When the mechanical aspects were incompletely mastered, the trainee expressed difficulty in simultaneously handling the more abstract problems of question formulation and search vocabulary negation; the designedly small file segments available to the emulators compounded the difficulty by allowing less chance of accidentally productive search statements.

This prompted our generation of a series of suggestions for "independent" practice on the emulator; a further benefit of the suggested searches was discovered in the process. It is possible, through careful selection of search terms and strategies of logical combination, to enhance and expand the concepts taught in the CALP modules, in addition to simple reinforcement. More work on the "suggestions" is planned when data base segments are fully developed.

The "suggestions" also prompt us to consider adding a further module to the training program: using results of trainee output for canned searches we will show him how his choice of terms and strategies affect the precision and recall values for his search results, and the cost of his search.

Section 2 S.T.I. Training Modules Report

December 1977

Computer Assisted Training and Practice Modules
Individual Primarily Responsible: Roger Rathburn*

Related Appendices:

C. Illustrations of Runs of CALP Modules I - VII, by trainees

D. Program listings of CALP Modules I - VII (for Modules VIII - IX, see Section 3 of this report)

---

*The very valuable assistance of Jean Guasco, in numberous "user" runs during the "debugging" and "bullet-proofing" processes, is gratefully acknowledged.

The Computer Assisted Learning and Practice (CALP) Modules

The Computer Assisted Learning and Practice (CALP) Modules form one major component of the Scientific and Technical Information (STI) services training package. There are nine modules. Each module covers a different segment of the total computer operations that the end user of STI services must learn in order to make intelligent use of a large, computer-based STI system, or an emulated system for non-punitive practice prior to use of the full-scale system. In this section we will discuss our reasons for using CALP, the goal(s) for its use, the rationale for specific ways we have organized the materials, our specific objectives, our progress to date, and some preliminary conclusions.

Reasons for Using CALP

There are a number of reasons why we used the CALP modules as part of the training package.

The first, and most obvious, reason is that of the subject matter being learned. The users of the training package are learning to use computer based, online information retrieval systems. It appears to us that the best way to learn the operations and commands involved is to actually practice operations and commands that are (as much as possible) exactly the same as those on the system(s). This can be done through the use of the CALP modules, without incurring the expense of large-scale data base processing or long distance communication costs.

In addition, the CALP modules guide the learner through his first experiences with a command or procedure, giving hints about how best to proceed, and help when it is requested. The actual STI search systems do not, as a rule, do this. The user is expected to know what to do as soon as he has logged on (if the user knows how to logon).

There is one more not-so-apparent reason for our use of CALP. This training package is meant to be used by the end user of the Scientific and

Technical Information; not by a professional search intermediary.* It is probable, then, that many of the users of this training package have never had any experience in interacting with a computer. These naive users can start using the CALP modules with almost no knowledge of how to interact with a computer, and build confidence in their new-found abilities in using a computer, along with competence in using one of the covered STI search services. The CALP module users will form realistic expectations of what computers can and can't do, and will not be hesitant to use them.

(*We have found, however, that professional intermediaries who are just beginning to use the services welcome this training opportunity as an effective, inexpensive way to initiate themselves, and later, to bring other untrained colleagues online)

GOALS

The goals that we hope to attain with the CALP modules are basically the same as those we hope to attain with the entire STI training package. They can really be inferred from our just-completed discussion of reasons for using the CALP modules.

First of all we want to train the individual to become a competent, relatively efficient, confident user of one or more of the large, computer-based STI search services. We mean that the user should be able to make a relatively straightforward basic search on at least one of the STI services in an efficient and competent manner. We do not mean that we expect the user to be an "expert". In general, such expertise only develops out of considerable actual experience in real"need to know" situations on the full-scale systems.*

In addition, we expect that the user will develop a good, sound, fundamental ability in the use of computer systems, per se; a basic understanding of how such systems can be used, and of what their capabilities and limitations are.

---

*See Appendix A, Levels of Competence

Implementation of the CALP Materials

The CALP modules are written in the CATALYST/PIL language. PIL (Pitt Interpretive Language) is a powerful, high level, conversational language developed at the University of Pittsburgh, while CATALYST is a specially designed program that enables one to enter textual teaching materials very efficiently. CATALYST is based on PIL and allows the programmer full use of PIL's extensive capabilities in a CATALYST lesson.

The languages are implemented on the University cf Pittsburgh's DEC system-10 computers.

The CATALYST/PIL language was used by us because of its ready availability, and because of its wide range of capabilities. In philosophy and implementation it belongs to the class of programming languages usually designated as "lesson-designer" languages.

Content and Organization of the CALP Materials

As was earlier stated, this training package is aimed at the beginning user of interactive computer systems, and is designed to bring that user to a basic level of competence for searching on at least one of the large STI service systems. This means that some of the areas covered are extremely basic (i.e., how to login to a system). It also means that we do not attempt to cover all the wide range of details that would make one an "expert" in the use of a system. What we included are the most commonly used (according to our findings and those of others that we consulted) commands and procedures for the given system being learned, along with any additional materials that we judged would be helpful in training the user in basic, efficient, competent searching of that system.

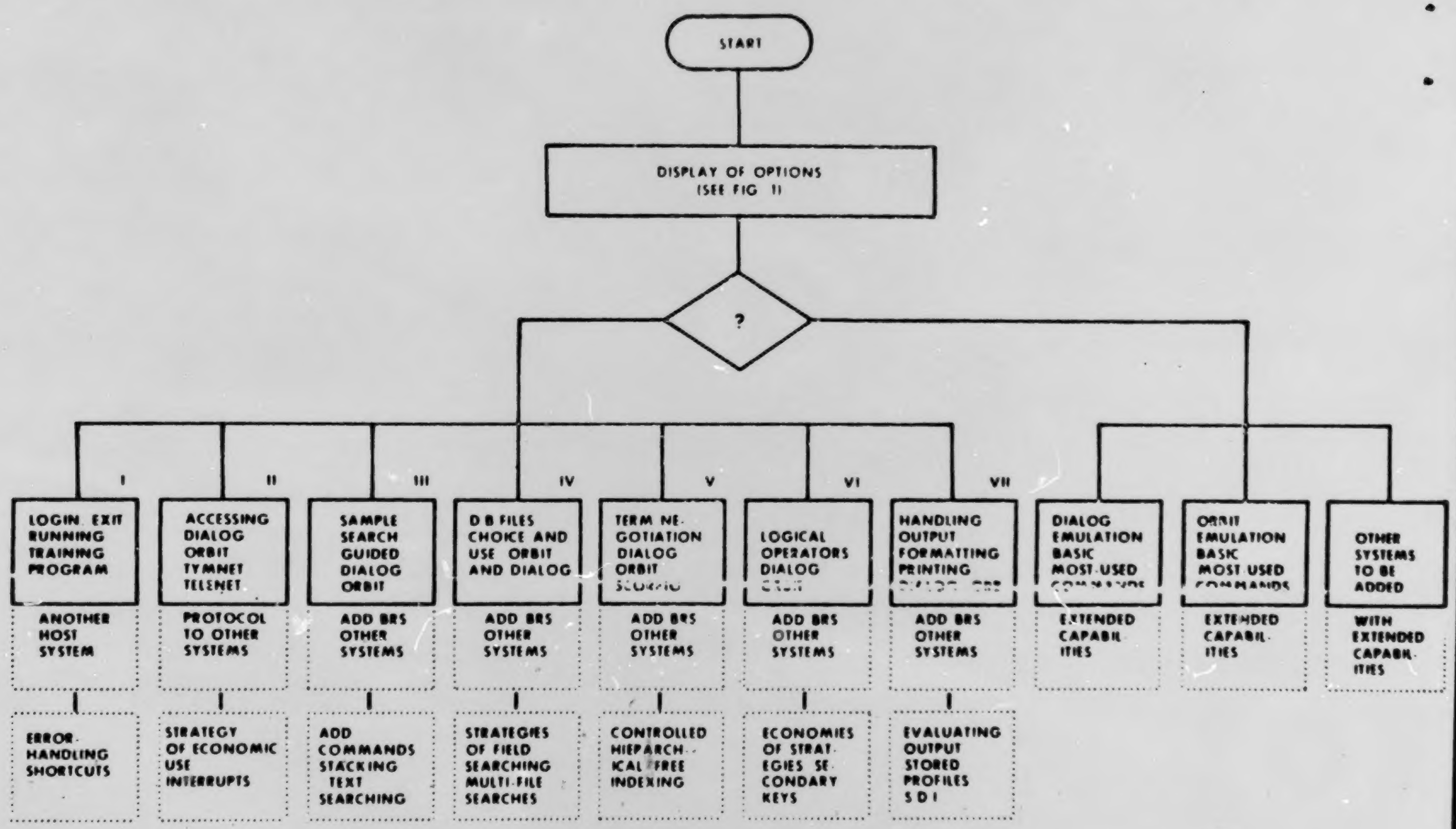
Once we had decided upon what materials should be included in the CALP package for each system, we could see that the amount was much too large to learn at one terminal session. We decided to break the materials down into seven "modules." Each module covers a basic functional area involved in conducting a complete search session. In the next section we will go into more detail about each of the modules.

The CALP Modules

Although the content of each module differs from that of each of the other modules, the general pattern for their use is similar. After selecting a given module, the user is first introduced to the general concepts and procedures included in the module. Then he is given a choice as to which system he wants to learn more about. After he has chosen a particular system, the specific commands and procedures for that system are presented to the user. Usually for each there is a brief explanation of the actions involved, then an example is presented. The user then runs through an exercise involving the use of the command or procedure. After completing the specific materials on one system, the user usually has the choice of practicing the comparable commands on another system, going to a system emulator, or leaving the CALP package.

This is the general pattern for the modules. The individual modules cover the following aspects of online searching (see Figures 1 and 2).

I. Introduction(Accessing the Pitt System and Modules)
II. Accessing the Search Service Networks
III. Search Practice
IV. Data Base Features
V. Negotiating Search Terms
VI. Use of Logical Operators
VII. Handling Search Results
VIII. Emulation of DIALOG
IX. Emulation of ORBIT



START
DISPLAY OF OPTIONS (SEE FIG 1)
?
I
II
III
IV
V
VI
VII
LOGIN EXIT RUNNING TRAINING PROGRAM
ANOTHER HOST SYSTEM
ERROR-HANDLING SHORTCUTS
ACCESSING DIALOG ORBIT TYMNET TELENET
PROTOCOL TO OTHER SYSTEMS
STRATEGY OF ECONOMIC USE INTERRUPTS
SAMPLE SEARCH GUIDED DIALOG ORBIT
ADD BRS OTHER SYSTEMS
ADD COMMANDS STACKING TEXT SEARCHING
D B FILES CHOICE AND USE ORBIT AND DIALOG
ADD BRS OTHER SYSTEMS
STRATEGIES OF FIELD SEARCHING MULTI-FILE SEARCHES
TERM NEGOTIATION DIALOG ORBIT SCORPIO
ADD BRS OTHER SYSTEMS
CONTROLLED HIERARCHICAL FREE INDEXING
LOGICAL OPERATORS DIALOG ORBIT
ADD BRS OTHER SYSTEMS
ECONOMIES OF STRATEGIES SECONDARY KEYS
HANDLING OUTPUT FORMATTING PRINTING DIALOG ORB
ADD BRS OTHER SYSTEMS
EVALUATING OUTPUT STORED PROFILES S D I
DIALOG EMULATION BASIC MOST-USED COMMANDS
EXTENDED CAPABILITIES
ORBIT EMULATION BASIC MOST USED COMMANDS
EXTENDED CAPABILITIES
OTHER SYSTEMS TO BE ADDED
WITH EXTENDED CAPABILITIES

Figure 2 TRAINING MODULES FOR USE OF S.T.I. SERVICES
AUGMENTATION AND EXTENSION OF SOFTWARE*

Existing Computer Assisted Learning and Practice Modules and Emulators (solid lines) will be augmented to include more systems, e.g., B.R.S., SCORPIO..... They will be extended to include more sophisticated capabilities, e.g., use of "stacked" commands, S.D.I. uses, use of secondary keys, field limiting, as provided in the "target" systems.

*Contents of blocks is indicative, not comprehensive.

Current State of the CALP Modules

The reader should now have a clear picture of the CALP portion of the STI training package. We have detailed our reasons for using CALP, and our plans for constructing CALP modules. Now we briefly describe where we presently are with the modules, and what we have found out so far about their use (see also "Use and Evaluation," p.62).

The DIALOG system is completely represented on all seven modules, and the materials on it have been fairly extensively used.

The ORBIT system is now represented on the seven modules. These materials have not really been extensively used yet, since they were not complete during the early testing period. The ORBIT emulator has just been completed; demand factors (DIALOG has been made a part of the University's Library's service) are responsible for this early focus on DIALOG. Several modules also contain material on the SCORPIO system, but we have not yet really worked on that system to any great degree.

Our organization of the modules has followed the pattern that we described earlier, and we have encountered no great problems in following it.

Conclusion

Our preliminary analysis of the use of the DIALOG materials has shown us that people are generally using the modules as we had anticipated, first using several of the early modules, then the emulator, and then back and forth between modules and emulator. Details about the evaluation of the CALP modules and the entire training package are presented in another section of this report, but it appears that we are, at least, on the right track (see"Use and Evaluation," p. 62).

We already anticipate making some changes in the modules that we have thus far implemented; but this was expected, and is the reason for our ongoing evaluative procedures. As the project progresses further, we will be better able to refine our approach.

The materials are flexible in use. The printed materials consist of a manual on the use of the training program itself and on DIALOG and ORBIT search commands, and a set of coordinated exercises to use with the computer assisted learning and practice modules. The manual exercises, and CALP modules could be used to take trainees directly into use of the online services; we have, however, developed the second level of training programs, the service emulators, which we believe will speed user skill development, and at less cost, in actual online dollar charges, and in inadequate search results. These are the topic of the next section of this report.

Section 3 S.T.I. Training Modules Report

December 1977

Emulation of DIALOG™ and ORBIT™ Retrieval Systems
Individual Primarily Responsible: John Griffiths

Related Appendices

E. Illustrative Runs of Emulators by Trainees

F. Program Listings for DIALOG and ORBIT Emulations

Fortran Source Programs and Flow Charts

I Introduction

II Retrieval System Emulation

III System Design

IV Programming

V Multiple System Emulation

VI Conclusion

VII Exhibits

  A. Functional Analysis of Command Routines - DIALOG and ORBIT

  B. DIALOG Commands Recognized

  C. DIALOG Commands Implemented

  D. DIALOG Error Messages

  E. ORBIT Commands Recognized

  F. ORBIT Commands Implemented

  G. ORBIT Error Messages

  H. Emulator Program System Cross-reference List

  I. Common Storage Map -- DIALOG Emulator

  J. Common Storage Map -- ORBIT Emulator

  K. File Cross-reference List

  L. DPB and LDB Library Routines

  M. Housekeeping Notes and Programs

  N. Notes on Limits

I. This report covers work done at the University of Pittsburgh from September 1976 through September 1977. The work involved programming a system to provide online terminal interaction similar to certain large-scale interactive bibliographic data retrieval systems.

Specifically, the programming mission can be described as creation of a system of programs capable of emulating basic aspects of Lockheed's DIALOG and SDC's ORBIT (emulate in the sense of imitating, rather than the sense of equalling or excelling). Vendor-supplied user manuals provided specifications of system behavior for emulation. The mission is analogous to the archaeological reconstruction of a prehistoric animal given only some fossilized bones. An advantage of this approach: we did not emulate vestigial functions!

II. Preliminary analysis of the user manuals revealed that both DIALOG and ORBIT are fundamentally alike inasmuch as they both provide a similar service in a similar mode through similar command language processing, search and retrieval operations, and display activities.

For purposes of emulation, the precise command language and display characteristics of DIALOG and ORBIT are critical; that is, each system has its own specific user-visible command language and display characteristics, but the processing which generates the displayed data, or which interprets and executes user commands may be, in its results, very similar. This observation leads to a basic design decision for emulator programming; that wherever possible, the same processing, search and retrieval procedures should be used for supporting the different command language and display activities. Thus, portions of the different emulator systems consist of identical programs for processing, search and retrieval operations.

III. The basis for the design of critical activity programs is provided by structural-functional analysis of command languages within the context of interactive operations. The operational characteristics of the emulator system are defined by the command language capabilities implemented. These capabilities are specified by the overall project goals, and are defined as: those capabilities with which a user should be familiar in order to have a "first level of competence"* in the use of DIALOG and ORBIT.

Structural analysis of the command languages showed that the emulator programs should support data base selection; search negotiation; search execution; output manipulation and examination; and session summary and termination.

Functional analysis of the command languages proceeded in two steps; first, the commands necessary for support of a satisfactory emulator structure were identified, and then functional analysis of those commands provided the design basis for programming the individual command support routines (See Exhibit A).

The interactive context provides a framework for analysis and an operational definition of emulator capability--what the system does is what the user sees. Therefore analysis of display characteristics--the system's response to user input--was necessary for complete specification of command support activities.

The design of the processing programs was also based upon structural-functional analysis, again within the context of interactive operations. Processing encompasses file management; command recognition; validation of user input; evaluation of logical expressions; short-term storage and retrieval of search and logical expression results; and error detection, diagnosis and recovery. The most critical processing activity is command recognition, especially since the command languages are grammatically ambiguous. That is, the languages are not based upon generative grammars which completely define their syntax.

---

*Defined Training Manual (Appendix A)

Therefore the best analysis of command languages which could be performed was a straight classification based upon semantic identity. One of the advantages of this classification is that it becomes a simple task to recognize all commands, although, as has been noted, not all command capabilities are emulated. This results in what has been called "user-cordial" interaction. A user may input any command and the system will determine if the command is implemented, not implemented, or invalid. Thus the distinction between invalid commands and non-implemented commands is maintained.

For example, in DIALOG, if a user enters BEGIN, the system emulates the DIALOG response. If a user enters SORT (a valid DIALOG command), the system responds with a "command not implemented" message; however, if the user enters RUN (not a valid DIALOG COMMAND) the system responds with an "invalid command" message.

The semantic identity of a command is the unique meaning associated with that command. For the user, the meaning of commands is learned through training; for the program system, meaning is associated with a unique code, or index, for each recognizable command. For example, in DIALOG, the commands E, ", and EXPAND have the same semantic identity (see Exhibits B and C).

IV. The primary consideration in the choice of programming language was transportability. That is, the emulator program system had to be written in a source language which could be compiled and executed by a variety of computer systems. Further, the language of choice had to be capable of handling online terminal and file operations. The language chosen for emulator programming was FORTRAN, and the system was developed on the University of Pittsburgh's DEC system-10 using the FORTRAN-10 compiler with FOROTS.*

The transportability consideration, of course, involved a tradeoff in program efficiency--FORTRAN does not support character-string manipulation. Since commands are entered as character strings, this represents a major programming problem. FORTRAN's powerful array-handling capabilities cannot be used directly for manipulating strings of characters since arrays are handled as words and characters are packed (in DEC FORTRAN) five to a word.

Therefore, the programs use bit-manipulation routines from the DEC-10 FORTRAN library, namely, subroutine DEPOSIT BYTE (DPB) and function LOAD BYTE (LDB), which move character-size bytes (see Exhibit L). When transporting the program to another environment, changes in character handling operations may be required depending upon:

- How many ASCII characters can be stored in a machine word
- What facilities for bit, byte, and character manipulation are supported

For example, IBM 360 series machines support a move character instruction which should be substituted for DPB and LDB operations, if the programs are transported to a 360 environment which includes an online FORTRAN operating system.

---

*Version 4 FORTRAN-10/Version 4 FOROTS (FORTRAN Object Time System) "The primary function of FOROTS is to act as a direct interface between user object programs and the DEC system-10 during input and output operations." DEC-10-LFORA-D-D, Nov. 1975.

For the emulator program string-handling is accomplished by using character-position indexing within arrays. The functions IWRD and IBIT are used to calculate, respectively, the word location and starting bit position of a single character within an array, given the index of that character.

For example, if in the LINE array the following 35-character string exists in the first seven words:

```
ABCDE.FGHIJ.KLMNO.PQRST.UVWXY.Z1234.56789
  1     2     3     4     5     6     7
```

then for any character position p, $(1 \leq p \leq 35)$ IWRD(p) gives the word and IBIT(p) the starting bit location of that character. Throughout the programs, maximum use is made of FORTRAN's array-handling feature, so that except for character-string operations, the programs are essentially table-driven.

The emulator programs access data base files in order to perform searches and to retrieve data (see Exhibit K). Except for data retrieval, which consists of direct file reading, the data base operations are functionally distinct from emulator operations. This distinction means that operations such as source record editing and formatting, and index building are performed by programs external to and separate from the emulator programs. From the point of view of programming tasks, the task of producing a program capable of searching indices for specific terms falls under data base operations. Therefore, the emulator system uses search routines supplied by the same programmer who created the other data base management programs. A call to a search routine produces a reference value which points to the file location of the index term passed as the calling argument.

For the emulator systems, the identification and validation of command language elements constitutes a critical operation--the operation upon which effective emulation depends. The process which results in validation of recognizable commands is implemented in a subroutine named VALID.

VALID is a multi-pass processor inasmuch as it examines input commands as many times as necessary in order to arrive at a unique identification of the input command. In the DIALOG emulator, for example, the first pass compares the first character input with the valid DIALOG single-character commands, and sets the unique index value of any recognizable match. However, some DIALOG commands may be valid either as a single-character abbreviation or a natural-language word; the significance of this lies in the fact that it is necessary to determine the location of the user-supplied argument which follows the command. Therefore, in the case of the command SELECT, which may be abbreviated S, recognition of the S must be followed by another pass which determines whether or not the characters ELECT follow the S, so that the location of the command argument, in this case an index term, may be determined.

Thus the multi-pass process allows the emulator to exhibit the same characteristics as the DIALOG system, even as far as anomalous "behavior". For example, the DIALOG user manual notes that selection of the term ELECTRIC may produce anomalous results as: the command abbreviation S, when used as SELECTRIC results in selection of the term RIC, since SELECT appears to be a natural language command. Similar results are obtained using the emulator:

INPUT	RESULT
SELECTELECTRIC	TERM ELECTRIC SELECTED
S ELECTRIC	TERM ELECTRIC SELECTED
SELECTRIC	TERM RIC SELECTED

ORBIT command words are identified and validated by the ORBIT-Emulator version of subroutine VALID, which processes all input lines which begin with the " character. It is relevant to note that the use, in ORBIT, of the " character is intended to designate the beginning of a string of characters. Although, strictly speaking, the emulators do not process strings, the program systems do apply string handling principles in the management of subprogram execution: a string is a type of *list*, and consists, in principle, of a sequence of parts, i.e., a command followed by an argument. The command processing subroutines, then, only process the argument part of an input string. The entire recognition and processing system can be represented very simply in list notation:

If ENTRY is an input list; and
if COMMAND is a function equal to the command string part of the input list; and
if ARGUMENT is a function equal to the argument string part of the input list; and
if VALID is a function which is equivalent to a process identified by a string; and
if PROCESS is a library of argument-string processing routines; then recognition and processing becomes the list operation:

((PROCESS(VALID(COMMAND(ENTRY))))(ARGUMENT(ENTRY)))

Command processing of arguments consists of identification and validation of terms, numbers, special characters, and logical operators, all in various combinations, depending upon the command input.

In addition to command recognition and processing, the emulator systems produce user-readable output at the terminal; and for the user, the emulator output is the critical feature of the system. Terminal output--the online operations of emulation--falls into *three areas of analysis for program design*. *First is format control*; the production of printed output, whether of search activities or results, which is formatted just like the output of the target system. The extensive format control features of FORTRAN allow exact reproduction of the printed output of target systems. The reproduction is based upon direct format analysis of examples and samples of target systems output.

Second, terminal output operations must conform to interaction cycle control; the printing of prompts and responses must be in the same sequence as those of the target systems. Interaction cycles are defined and programmed upon the basis of direct cycle analysis of examples and samples of target systems output.

Third, the error messages generated by the emulator systems must conform, both in format and diagnosis with the target systems. Definition of error messages and diagnosis of error conditions is based upon analysis of target systems manuals, and upon functional analysis of commands.

Thus, the emulator systems are basically identical, at least in interactive appearance and behavior to the simpler aspects of the target systems.

V. The emulator program systems are designed to facilitate expansion to cover other online search service systems. A new target system can be prepared for emulation by the same methods used in development of the emulator programs. These methods have been described earlier in this report, and consist of:

- Command Language Analysis
  - Syntax/Semantics
  - Function Analysis
- Format analysis of examples and samples of target system output
- Interactive cycle analysis of target system
- Comparative evaluation of functional capabilities already implemented with those required for new emulation

- Program changes
  - Command word tables
  - Format statements
  - Interaction cycle management
- Program additions and deletions

VI. In conclusion, this report has shown how a system of FORTRAN programs for emulation of major online search systems was developed. The problem solving approach was guided by principles of general systems analysis; information retrieval system design; linguistic analysis; list structure processing; and block-structured programming.

Perhaps the most innovative aspect of this work was the combination of linguistic analysis, list processing, and block structured programming. During program execution, the derivation of a unique syntactic identity, from an input string, generates the semantic or logical state necessary for correct execution of the appropriate segments of the FORTRAN program system. Of course, this statement implies, as has been stated earlier in this report, that command words actually have some meaning to the emulator program system. The only justification for this use of the notion "meaning" is that the emulator system behaves appropriately, as if the meaning were comprehended; this is of course our pragmatic test for the apprehension of the meaning of a command in more complex (human) systems as well. What actually happens is that as the program system identifies an input string, the "compare-decide" process process actually sets pointers and flags; changes control values; and acquires the storage necessary for the subsequent execution of the program segments associated with the input string.

This method has at least two operational advantages: first, the elimination of inappropriate system behavior during emulation--if an input string does not express necessary and sufficient semantic content, the system will not attempt to execute; and second, the facilitation of error diagnosis during program execution--the state of pointers and flags, control values, and the contents of assigned storage are all examined during error checking, which incidentally, is the highest priority program activity.

Thus, grammatical analysis of command language systems provides operational specification of emulator program structures, and further, provides comparative specifications for minimizing redundancy in emulator programs.

Exhibit A-1

DIALOG FUNCTIONAL ANALYSIS OF COMMAND ROUTINES

1. BEGIN ! B (B)

Overview

BEGIN identifies the user by TITLE, SEARCHER, and REQUESTOR, and acquires a reference number for the file to be accessed.

The identification section can be bypassed and the file reference number is set to the default value of 1.

Error Conditions

An invalid file reference number will generate the following message:

INVALID COLLECTION NUMBER

The BEGIN routine is then restarted.

See Flowchart 1

2. EXPAND " E (TERM) (E#)

Overview

The EXPAND command operates in two modes. The first mode is specified by entry of a search term, the second by entry of an E-number.

Mode 1 " TERM, E TERM

Entry of an EXPAND command followed by a search term causes a listing of terms that are alphabetically close to the entered term (5 before, 9 after) to be printed.

Mode 2 " E#, EE#

Entry of an EXPAND command followed by an E-number causes a listing of terms related to a term listed by a previous mode 1 EXPAND command.

Mode 2 is only valid if the requested E-# has a number of related terms indicated in the previous Mode 1 listing.

Error Conditions

1. An attempt to EXPAND a term that is not in the current file index will generate the following message:

Exhibit A-2

TERM NOT IN DICTIONARY

The EXPAND routine is aborted.

2. An attempt to EXPAND related terms before a listing of alphabetically close terms has been generated will generate the following message:

NO DISPLAY FOR ITEM

The EXPAND routine is aborted.

3. An attempt to EXPAND related terms for an E-number which is less than 1 or more than 16 will generate the following message:

INVALID ARGUMENT

The EXPAND routine is aborted

See Flowchart 2

3. SELECT # S (TERM) (E#)

Overview

The SELECT command causes the citations indexed by the TERM OR E-number to be identified with the current SET NUMBER for future reference. SELECT operates in two modes.

Mode 1 (direct)

A SELECT command followed by a term causes an index search for that term, and all 'hits' are identified as members of the current SET.

Mode 2 (indirect)

A SELECT command followed by an E-number causes all citations associated with the E-number to be identified as members of the current SET. A Mode 2 SELECT is only valid if a previous EXPAND command has generated a list of E-reference numbers.

SELECT will process multiple E-numbers, using OR logic, either individually-entered separated by commas, or inclusively-range indicated by a dash, i.e., E_n-E_{n+i}

where: $1 \leq n < n+i \leq 16$

Error Conditions

1. An attempt to SELECT a term that is not in the current file index will generate the following message:

TERM NOT IN DICTIONARY

The SELECT routine is aborted

2. An attempt to SELECT a related term using an E-number before an EXPAND command has generated a reference list will generate the following message:

NO DISPLAY FOR ITEM

The SELECT routine is aborted.

3. An attempt to SELECT a related term using an E-number that is less than 1 or greater than 15 will generate the following message:

INVALID ARGUMENT

The SELECT routine is aborted.

4. An attempt to SELECT terms that constitute a 51st SET will generate the following message:

50 SETS ALREADY GENERATED

The SELECT routine is aborted.

See Flowchart 3

Exhibit A-4

4. COMBINE $ C

Overview

COMBINE or C command is used to perform Boolean operations, AND, OR, NOT on sets created by the SELECT command. Only set numbers can be used in COMBINE statements. By using parentheses, a number of COMBINE operations can be accomplished in one command.

The result of the COMBINE command is a set number with a number of postings.

Error Conditions

An attempt to enter a COMBINE command without an argument generates the following error message:

EMPTY ARGUMENT STRING

An attempt to COMBINE an expression containing more than 25 setnumbers (either specified or implied, as in a range) generates the following error message:

BOOLEAN STRING TOO LONG

An attempt to COMBINE an expression containing an invalid setnumber generates the following error message:

INVALID SET VALUE

An attempt to COMBINE an expression containing an unequal number of left and right parentheses generates the following error message:

PARENTHESES MISMATCH

An attempt to COMBINE an expression with a misplaced "/" operator generates the following error message:

INVALID IMPLICIT COMMAND

An attempt to COMBINE an expression that is not syntactically correct generates the following error message:

ARGUMENT SYNTAX ERROR

An attempt to COMBINE an expression which evaluates equal to an empty set generates the following error message:

SET HAS NO MEMBERS

Exhibit A-5

If the program encounters an error not caused by the user, the following error message is generated:

UNKNOWN ERROR CONDITION

If the program encounters a file error during execution, the following error message is generated.

FILE ERROR

In the event of any error, control is returned to the monitor program.

5. EXPLAIN FILES ?FILES

Overview

EXPLAIN FILES or ?FILES is used to get an updated listing of the data bases available.

6. TYPE T

Overview

TYPE or T command is used to type the previously retrieved sets on the terminal. The following is a general format for the TYPE command:

TYPE SET #/FORMAT NO./NUMBER OF ITEM OR RANGE OF ITEMS REQUESTED

Error Conditions

An attempt to use the TYPE command before the SELECT command will generate the following message:

```
NO SETS CREATED
```

An attempt not to type the set number,or to type a set number which does not exist, will generate the following message:

```
INVALID SET VALUE
```

An attempt to exceed the existing available format codes will generate the following message:

```
INVALID FORMAT CODE
```

Any syntactic error in the argument will generate the following message:

```
ARGUMENT SYNTAX ERROR
```

An attempt to exceed the item reference number will generate the following message:

```
ITEM REFERENCE NUMBER (E#) OUT OF BOUNDS
```

7. DISPLAY SET DS

Overview

DISPLAY SETS or DS allows the user to see a summary of the search strategy at any time during the online session.

Error Conditions

A typographical error will generate the following message:

```
COMMAND NOT CURRENTLY SUPPORTED
```

Exhibit A-7

8. .FILE(n)

Overview

This command is used for initial file selection or change during a search. It does not reset the set number counter or erase the previous search history as the BEGIN command does.

When this command is used, the system responds with a search summary and file update statement.

Repeated use of .FILEn can result in exceeding the system's capacity of 50 sets.

Error Conditions

Any typographical error or abbreviation will generate the following message:

```
INVALID COMMAND
```

9. END

Overview

This command terminates the search and summarizes the online activities. It does not erase the previous search.

With this command, the system responds with date, time, name of the user, cost and time spent in file.

10. LOGOFF

The LOGOFF command puts an end to the online activities. It indicates name of the user, cost and time in hours spent in file.

Exhibit A-8

SDC/ORBIT COMMAND FUNCTION ANALYSIS

SEARCH STATEMENT (Default Response)

1. Normal (when SSn is successful)

PROG:
SSn Pstg (# of postings)

2. Abnormal (when search is not successful)

PROG:
NP (term)
*NONE-

3. Error (when SS contains a mistake, i.e., n and term entered when $<$n SS's exist)

PROG:
? (Error Message)

NOTES

All default mode entries have to be parsed for terms and keywords: OR, AND, AND NOT

Logical operator priorities: AND $<$ AND NOT $<$ OR

FIND COMMAND

"FIND or "FD

Forces search statement processing of field following command. Response is as if a SS had been entered. See SEARCH STATEMENT.

HISTORY COMMAND

"HISTORY

Defaults to a listing of all SS's, unless a valid SS# follows the command, when just that SS description is printed.

PROG:
SS n: Line as entered (# of postings)
SS n_i: Line as entered (# of postings)
SS n_j: Line as entered (# of postings)

NEIGHBOR COMMAND

"NBR

Displays alphabetically close terms (a "window") from index. Enter a sub-interaction to move the "window" up or down the index list.

PROG:
Postings Term
(5-item list)
UP N or DOWN N?

Exhibit A-9

PRINT COMMAND

```
"PRINT   "PRT   "P
```

The "PRINT command defaults to a listing of key bibliographic categories for up to 5 citations most recently retrieved.

The "PRINT FULL form produces a full listing of the last record retrieved.

The "PRINT TRIAL form produces listings of subject-related categories from the last two (2) citations retrieved.

Any valid SS reference number may be entered to force the PRINT to use the citation set associated with that SS.
i.e.,

```
USER:
"PRINT SS 1
```

Any number of citations may be requested in "PRINT by entering the desired number.
i.e.,

```
PROG:
SS 1 PSTG (200)

USER:
"PRINT 20
```

The full (FULL) format includes the two-letter designators for printable fields.

EXCLUDE and SKIP instructions will be recognized in 3rd priority implementation.

STOP COMMAND

```
"STOP
```

ORBIT asks the user to verify the STOP, prints a time, online activity summary, and disconnects.
i.e.,

```
USER:
"STOP

PROG:
DONE? (Y/N)

USER:
Y

PROG:
TERMINAL SESSION FINISHED DD/MM/YY   HH:MIN A/P.M (PACIFIC TIME)
ELAPSED TIME ON (filename): 0.00 HRS.
TOTAL ELAPSED TIME:  0.00 HRS.
```

Exhibit B

DIALOG Commands Recognized by the Emulator

COMMAND	FIELD LENGTH	INDEX = COND	ABBREVIATION	SYMBOL
(INVALID	N/A	0	N/A	N/A)
END	3	1	N/A	=
TYPE	4	2	T	'
PAGE	4	3	P	0
BEGIN	5	4	N/A	!
PRINT	5	5	* PR	&
LIMIT	5	6	L	)
EXPAND	6	7	E	"
SELECT	6	8	S	#
LOGOFF	6	9	N/A	N/A
COMBINE	7	10	C	$
DISPLAY	7	11	D	%
EXPLAIN	7	12	N/A	?
LIMITALL	N/A	13	* LALL	*)ALL
DISPLAY SETS	N/A	14	* DS	@
.(FILE)	4	15	N/A	N/A
.(SORT)	4	16	N/A	N/A
.(RECALL)	6	17	N/A	N/A
.(EXECUTE)	7	18	N/A	N/A
.(RELEASE)	7	19	N/A	N/A

Exhibit C.

DIALOG Commands Implemented in the Emulator

COMMAND	LENGTH	INDEX = COND	ABBREVIATION	SYMBOL
INVALID				
END	3	1	N/A	=
TYPE	4	2	T	,
BEGIN	5	4	N/A	!
PRINT	5	5	* PR	&
EXPAND	6	7	E	"
SELECT	6	8	S	#
LOGOFF	6	9	N/A	N/A
COMBINE	7	10	C	$
EXPLAIN	7	12	N/A	?
DISPLAY SETS	N/A	14	* DS	@
.(FILE)	4	15	N/A	N/A

Exhibit D

DIALOG

INDEX	ERROR MESSAGE
1	INVALID ARGUMENT
2	EMPTY ARGUMENT STRING
3	(INTER-SUBROUTINE CONTROL CONDITION)
4	NO PREVIOUS DISPLAY - ITEM REQUEST NOT SERVICEABLE
5	ITEM REFERENCE NUMBER OUT OF BOUNDS
6	50 SETS ALREADY GENERATED
7	BOOLEAN STRING TOO LONG
8	INVALID SET VALUE
9	PARENTHESES MISMATCH
10	SET NUMBER - SYNTAX ERROR
11	INVALID SET-RANGE OPERATOR
12	INVALID FILE NUMBER
13	INVALID FORMAT CODE
14	INVALID IMPLICIT COMMAND
15	ARGUMENT SYNTAX ERROR
16	UNKNOWN ERROR CONDITION
17	THESAURUS NOT ON LINE - RELATED TERMS NOT FOUND
18	COMMAND NOT CURRENTLY SUPPORTED
19	FILE ERROR
20	SET FULL
21	NO SETS CREATED
22	SET HAS NO MEMBERS
23	AUTHOR NAME FOUND
24	AUTHOR NAME NOT FOUND

Exhibit E

SDC/ORBIT Commands Recognized by the Emulator

COMMAND	FIELD LENGTH	INDEX = COND	ABBREVIATIONS
(INVALID	N/A	0	N/A)
NBR	3	1	N/A
FILE	4	2	(FILES?)
FIND	4	3	FD
HELP	4	4	N/A
KEEP	4	5	N/A
NEWS	4	6	N/A
STOP	4	7	N/A
TIME	4	8	N/A
ORDER	5	9	N/A
PRINT	5	10	P PRT
BACKUP	6	11	N/A
RENAME	6	12	N/A
STORAD	6	13	N/A
VERSION	7	14	VERS
DIAGRAM	7	15	DIAG
COMMENT	7	16	N/A
EXPLAIN	7	17	N/A
HISTORY	7	18	N/A
RESTART	7	19	N/A
ERASEALL	8	20	N/A
SECURITY	8	21	N/A
SUBHEADINGS	10 (11)	22	SUBS

Exhibit F

SDC/ORBIT Commands Implemented in the Emulator

" COMMAND	FIELD LENGTH	INDEX = CON	ABBREVIATIONS
NBR	3	1	N/A
FND	3(4)	3	FD
STOP	4	7	N/A
PRINT	5	10	P PRT
HISTORY	7	18	N/A
FILE		4	N/A
(FILES?		6	N/A

For ORBIT, any entry that is not a valid command and which does not begin with the " character, is treated as a SEARCH STATEMENT (SS).

SS's can consist of search terms, logical operators (AND, OR, AND NOT), and numerical references to previous SS's.

SS's therefore perform both searching and logical manipulations of search results.

Exhibit G

ORBIT

INDEX	ERROR MESSAGE
1	IS NOT A CORRECT COMMAND NAME
2	MISSING DOUBLE QUOTE MARK
3	PLEASE SEE CAI MODULES - TYPE "STOP THEN Y TO EXIT
4	COMMAND NOT SUPPORTED
5	INVALID SEARCH STATEMENT REFERENCE NUMBER
6	SEARCH SERIES FULL, ENTER COMMAND
7	SEARCH STATEMENT CONTAINS TOO MANY TERMS
8	ILLOGICAL STATEMENT ENTERED
9	INVALID SPECIFICATION
10	FILE NOT FOUND - USE "FILES?
• • • • 15	NOT USED
16	PROGRAM EXECUTION ERROR
17	ORBIT FILE ERROR
18	UNKNOWN ERROR CONDITION

Exhibit H-1

Program Cross-Reference

PROGRAM TASK	ORBIT(name)	DIALOG(name)
Command Input	ENTRY	ENTRY
Command Validation	VALID	VALID
End Search	(in MONITOR)	DEND
Online Printing of Search Results	OPRINT	DTYP
Crt Display	*	DISPL
Explain System	*	EXPL
Set Standard Limits for References	*	LMTAL
Starting Search After Login	N/A	BEGIN
CRT Page	*	PAGE
Print Search Results Offline	*	DPRINT
Set Limits for References	*	LIMIT
Expand Index Term	NBRTRM	EXPAND
Perform Logical Manipulation of sets	(in SRSTAT)	COMBIN
Disconnecting from System	(in MONITOR)	LOGOFF
Create Sets of References	SRSTAT	SELECT
Type Set History	SSHIST	DYSET
Search Statement Analysis and Result Posting	SRSTAT	———
Stack Entry	———	PUSH
Logical Manipulation of Sets with AND	CAND	CAND
Logical Manipulation of Sets with OR	COR	COR
Logical Manipulation of Sets with NOT	CNOT	CNOT

---

*Inactive

Exhibit H-2

PROGRAM TASK	ORBIT(name)	DIALOG(name)
Search Index for Term	SEARCH	SEARCH
Search Index for Truncated Term	TSRCH	TSRCH
Construct Character String	MAKE	MAKE
Set String Pointers for Source Reference Levels	REFSET	REFSET
Pack Arrays	PACK	—
Find Character in Command String	ICHAR	NCHAR
Find Next Character in Command String	NCHAR	—
Word Indexing For Strings	TWRD	IWRD
Bit Indexing for Strings	IBIT	IBIT
Find Special Characters in Command String	NXT	NXT
Remove Top Entry from Stack	—	IPOP
Set Operator Precedence Level	—	LEV
Counts Characters in Line Strings	ICOUNT	—
Set Source Field Level Reference Level Index	IVAL	—

Exhibit I-1

Common Map of Common Storage of DIALOG Emulator Programs

NAME	TYPE	SIZE(words)	PURPOSE
LINE	Integer Array	12	Input Line Buffer Contains commands and arguments
COND	Integer	1	Inter-subroutine Control Code
P	Integer	1	Character Position Pointer for the Line Array
F	Integer	1	Active File Number
PASS	Integer	1	User Password
D	Real	2	Date
FTIM	Real	1	Time-in-File
DIRNAM	Double Precision	2	Directory File Name
DATNAM	Double Precision	2	Data Source File Name
AUNAM	Double Precision	2	Author Index File Name
DESCRN	Integer	4	File Descriptor Field
SETN	Integer	1	Set Number Counter
EPLACE	Integer Array	17	E-Number Reference Table
AUTH	Logical	1	Author Flag

Map of Labeled Common

Subroutines: COMBIN, PUSH, CAND, COR, CNOT FUNCTIONS: POP
LABEL: COMB

STACK	Integer Array	100	Postfix Holding Array in COMBIN, PUSH, POP
TOP	Integer	1	Stack Pointer in COMBIN, PUSH, POP
ISET	Integer	1	Work Set Number in CAND, COR, CNOT

Subroutines: CAND, COR, CNOT
LABEL: COMB

NAME	TYPE	SIZE(words)	PURPOSE
DD	Integer	101	Dummy Field Masks Stack and Pointer in CAND,COR,CNOT
TEMP	Integer	1	Result Workset in CAND, COR, CNOT; Equivalent to ISET

Subroutines: REFSET
LABEL: TYP

NAME	TYPE	SIZE(words)	PURPOSE
BWRD	Integer	1	Beginning Word of Field in Source Record
EWRD	Integer	1	Ending Word in Source
LEVEL	Integer	1	Source Field Level
IDIR	Integer Two Dimensional Array	40	Source Record Directory

Exhibit J-1

Common Map of Common Storage of SDC/ORBIT Emulator Programs

NAME	TYPE	SIZE(words)	PURPOSE
LINE	Integer Array	12	Input Line Buffer Contains Commands and Arguments
COND	Integer	1	Inter-Subroutine Control Code
P	Integer	1	Buffer Character Reference
SS	Integer	1	Search Statement Counter
IDUMMY	Integer Array	30	DUMMY

Map of Labeled Common

Subroutines: OPRINT, REFSET
LABEL: PRNT

BWRD	Integer	1	Beginning Word of Field in Source Record
EWRD	Integer	1	Ending Word of Field in Source Record
LEVEL	Integer	1	Source Field Level
IDIR	Integer Two Dimensional Array	40	Source Record Directory

Subroutine: SRSTAT
LABEL: COMB

STABL	Integer Array	10	Table for Constructing Logical Expressions
SSWRK	Integer	1	Working Set Number

Subroutine: COMBINE
LABEL: COMB

TABLE	Integer Array	10	Table of Entries Equivalent to STABL
SSWRK	Integer	1	Working Stet Number

Exhibit J-2

Subroutines: CAND, COR, CNOT
LABEL: COMB

NAME	TITLE	SIZE(words)	PURPOSE
DD	Integer	10	Dummy Field Printer in CAND, COR,CNOT
TEMP	Integer	1	Result Work Set in CAND,COR,CNOT Equivalent to SSWRK

Exhibit K

File Access Cross-Reference

FORTRAN SOURCE FILES	DIALOG	ORBIT
Monitor	DMON.F10	ORMON.F10
Subroutines	SDMON.F10	SORMON.F10
Functions	FDMON.F10	FORMON.F10

DATA FILES	UNIT #	DIALOG	ORBIT
Error Messages	21	ERRMS.DAT	ORBERR.DAT
File Specifications	29	DSPECS.DAT	OSPECS.DAT
Work File	28	SETWRK.DAT	SETWRD.DAT
Author Index	23	*	None
General Index	22	*	*
Documents Data	25	*	*

*As Read from Specifications File:

Documents	Data	Index	Author Index
Paper Chemistry	CHEM.DAT	CHEMIN.DAT	CHEMA.DAT
N.T.I.S.	NTISD.DAT	NTISIN.DAT	NTISA.DAT
Sociological Abstracts	SOCOL.DAT	SOCIN.DAT	SOCOLA.DAT
Engineering Index	CMPNDX.DOC	CMPNDX.IDX	CMPNDX.ADX

Exhibit L-1

University of Pittsburgh
DEC-10 Documentation-2
September 1975

LDB

LDB (LoaD Byte) is an integer function that extracts a byte of any length from any position in a word. By defining a byte to be 7 bits, LDB can be used to manipulate ASCII characters, which are each seven bits long.

Calling Sequence:

```
LDB (startingbit, bytelength, word)
```

"Startingbit" is an integer from 0 (left-most bit) through 35 (right-most bit); "bytelength" is an integer from 1 through 36; and "word" is any variable or constant.

LDB extracts a byte of "bytelength" bits from "word", beginning with the "startingbit". The function returns an integer result containing those bits, right justified, padded on the left with zeros. The contents of "word" is unchanged.

Example:

```
I=LDB(14,7,J)
```

Seven bits, beginning with bit 14, are moved from J into I, right justified and padded on the left with zeros. J is unchanged.

Exhibit L-2

DPB

DPB (DePosit Byte) is a subroutine that deposits a byte of any length into any position in a word. By defining a byte to be 7 bits, DPB can be used to manipulate ASCII characters, which are each seven bits long.

Calling Sequence:

```
CALL DPB (startingbit, bytelength, result, word)
```

"Startingbit" is an integer from 0 (left-most bit) through 35 (right-most bit); "bytelength" is an integer from 1 through 36; "result" is any variable; and "word" is any variable or constant.

DPB extracts a byte of "bytelength" bits from the right-most end of "word". These bits are placed into "result", beginning with the specified "startingbit". All other bits in "result" remain unchanged. The contents of "word" is unchanged.

By specifying a call to LDB as the fourth argument of a call to DPB, bits can be moved from any position in one word to any position in the same or another word.

Examples:

```
CALL DPB(12,4,I,J)
```

The right-most four bits of J are moved into I, beginning with bit 12, Hence, bits 12 through 15 of I will be set the same as bits 32 through 35 of J, and bits 0 through 11 and 16 through 35 of I will be the same as they previously were. J is unchanged.

```
DATA J/'QRSTU'/
CALL DPB(7,14,J,LDB(0,14,'ABCDE'))
```

Fourteen bits, beginning with bit 0, are moved from the literal constant 'ABCDE' into J, beginning with bit 7. All other bits of J will be the same as they previously were. Hence, J will contain the string 'QABTU'.

Exhibit M

Housekeeping

The specifications files contain the file names of the document data files and associated indexes. The specifications files are maintained by the interactive programs: DFILES.F10, for DIALOG, and OFILES.F10, for ORBIT. These programs contain complete instructions for specification of data and index files.

The specifications file is read by the emulator monitor at the beginning of execution, and the data and index files specified in the specifications file at that time become uniquely referenced by emulator file numbers for the duration of execution. File specifications cannot be modified during emulation.

The error messages files contain the text of error messages for the emulator. Error messages are indexed by error code (see exhibits D and G). Error message files are maintained by the interaction programs MSFILE.F10 and MSREAD.F10, for DIALOG, and MSORB.F10 and ERREAD.F10, for ORBIT. MSFILE.F10 and MSORB.F10 are programs which create, change, or delete the text of error messages, for DIALOG and ORBIT respectively. MSREAD.F10 and ERREAD are programs which only read the error message files for DIALOG and ORBIT, respectively.

Exhibit N

Notes on Limits

Apart from the limits imposed by non-implementation of commands (see exhibits B, C, and E, F), the emulator systems have other limitations as follows:

- There is no provision for supporting system news and messages;
- The emulators have no personal message facility, and do not support delayed search and output operations. Emulation is a connect-time only activity--no jobs exist except during emulation;
- The emulators do not support stored profile maintenance or SDI services;
- The emulators do not support an internal billing or cost estimation facility;
- During emulation, no more than 50 search sets in DIALOG, 25 in ORBIT, may be active at any one time; there are never more than 10 references per set; and search terms are limited to 20 characters.

Section 4 S.T.I. Training Modules Report

December 1977

Data Base Structure and Organization
Individual Primarily Responsible: James Williams

Related Appendices

G. Program Listings for File Generation and Maintenance

DATA BASES and THE NSF TRAINING PROJECT

I. Introduction

The NSF Training Project places a heavy emphasis on the mechanics and capabilities of the two major search systems, namely, Lockheed's DIALOG and SDC's ORBIT, but an equally important aspect of these search systems are the data bases that they manipulate.

In order to provide the user with data bases for searching purposes that emulate to some degree those available on the DIALOG and ORBIT systems, a series of steps were taken as outlined below.

II. Acquisition of Sample Data Bases

Over 60 data base vendors were contacted by the Office of Communications Programs and requested to supply a single issue of their data base as it would be supplied to DIALOG or ORBIT.

The response from the vendors was quite good and the most heavily used data bases on the DIALOG and ORBIT system were received. These include Chemical Abstracts, NTIS, Engineering Index, Paper Chemistry, Sociological Abstracts, Psychological Abstracts, and others (See Appendix A).

These data bases arrived in the format that the vendor normally distributes them. There is very little similarity among the data base formats. This led to step III which requires preprocessing of each source file.

III. Reformatting the Commercial Data Bases Into a Standard Internal Format

Because the data bases each have their own format and peculiar semantic and syntactical properties, it was not possible to derive a search program for the emulators that would handle all the differences without unreasonable overhead. Therefore a file structure was designed for the emulators that would provide for effective and efficient processing. The human factors constraints of response time and display formats were respected, while at the same time we attempted to conserve on computer resources, namely disk space and processing time. Therefore, an inverted file structure was designed with the following format:

20 bytes	1 word	1 word		1 word	1 word
Index Term	Level	Disk Address of Document	. . .	Level	Disk Address of Document
	1st Document			10th Document	

where:

Index Term - is a word or phrase up to 20 characters

Level - indicates where the index term for the document was taken from, e.g., author, title, keyword, abstract

Disk Address of Document - The Relative Record Number, where the document with the associated index term resides

A decision was made to permit up to 10 documents per index term. This was done to conserve disk space. It was also determined that 10 documents per index term for a single issue of a data base would satisfy 75% of all cases.

The index records are 120 bytes of 5 words (PDP-10) each. Thus 4.5 index records fit into a single disk block. Even this conservative approach requires a thousand plus disk blocks for a single issue of a data base.

The programming system to reformat the various data bases and build the related indexes requires four (4) programs. The first program reads the vendor formatted file and produces an intermediate format that is generalized in nature so that data elements in the file can be identified and in some cases edited to meet the DIALOG and/or ORBIT requirements. This intermediate format is shown below:

```
[Level     Data Element
```

Example:

```
[1  This is a title data element for use
[1  with the DIALOG/ORBIT emulators
```

Each record in the intermediate format begins with the "[" and is followed by a standard level code that identifies the data element. In the example above, the level code of "1" identifies the data element as a "Title." A record may not exceed 80 bytes but there may be any number of records for a single data element and up to 40 data elements per document. The last record for a document has a "#" in position one of the record to indicate the end of the document data elements. There is a separate reformat program for each data base since each has a different format.

Once the intermediate format has been produced, a program named "PIRCVT" is executed which performs two basic tasks. The first task is to format the data in the document for the DIALOG and ORBIT emulators and secondly to index the document based upon the data elements to be indexed and the type of indexing required. This program uses two tables to direct its processing related to indexing. Table 1 is the indexing guide table with the following format:

Level	Indexing Mode	Delimiter
0	0	0
1	2	0
2	1	;
3	0	0
4	1	;
5	0	0
6	0	0
7	0	0

Level refers to the data element type, e.g., 1 = title, 2 = author, 4 = keywords.

Indexing mode refers to: 0 = no indexing, 1 = entire data element indexing and 2 = word by word indexing (uniterm).

Delimiter refers to those elements that are to be indexed on a phrase by phrase on data element by data element basis. It informs the program when more than one occurrence of a data element exists, when the end of one has been reached and the beginning of the next one has been encountered. For example:

[2 Jones, Peter M; Smith, John J

The ";" indicates that the end of the first occurrence of the author has been encountered and another occurrence of an author follows. This is a convention produced by the program that generates the intermediate format. Thus, if the indexing mode was "1" (data element), the above record would generate two index entries, one for Jones, Peter M., and one for Smith, John J. This same schema holds for other data elements as well.

If the indexing is word by word, then a second table of "stop" words are utilized to eliminate certain words from appearing in the index. The "stop" word list contains about 40 entries, such as: a, an, and, the, of, about, etc.

The output from the "PIRCVT" program is two (2) files. One file contains the index terms and their associated document relative disk address. The file format is:

20 bytes	1 word	1 word
Index Term	Level	Disk Address

The second file is a file containing the document data elements as well as a directory of the levels and beginning addresses in the record of each data element. The format is:

Directory to Data Elements					Data Elements		
Level	Beginning Position	Level	Beginning Position	. . .	Data Element	Data Element	. . .

The Directory is used to support all of the variations of format used in the TYPE, PRINT, and DISPLAY commands allowed by DIALOG and ORBIT. The level indicates the data element type and the beginning position is a displacement from the beginning of the record indicating the starting position in the record of the data element.

IV. Editing of the Preliminary Index File

The index produced by "PIRCVT" must be edited in most cases to produce an index compatible with DIALOG and/or ORBIT. Certain marks of punctuation must be deleted and/or inserted to meet the specifications of thesauri or editing done by Lockheed and ORBIT.

This is done by using either the editor program on the computer system or a specially written program called "TRIM." In either case, the result is a preliminary index with the proper format and punctuation.

V. Building the Index

Once the preliminary index has been edited, it is sorted on Disk Address within Index Term to produce a file in alphabetically ascending order. This sorted file is utilized as input to a program named "INDEX" which reads each record in the preliminary index and builds a single record for each unique index term with the level indicators and document disk addresses. The format of this file was given previously. The result is an index file that corresponds to an "inverted" file structure. This permits rapid searches as well as data in a format amenable to "AND", "OR", and "NOT" logic combinations. Its weakness, of not being easily updated, is not relevant to the training project's utilization of the files.

VI. Program Characteristics

The programs that perform the indexing, editing, and index file build are written in ANSI FORTRAN. With very few modifications resulting from computer architecture characteristics, these programs can be run on other computer systems supporting ANSI FORTRAN with sequential and random access read and write capabilities.

A subroutine to pack and unpack characters into and out of words is the only system-dependent software that is included. This is only used to help conserve computer resources and a slight modification could eliminate the need for this routine. By the same token, a machine with byte addressable capabilities could eliminate the need for such a routine altogether. Thus, the programs are about as portable as they could be made, with very little machine dependency.

VII. Computer Aided Learning (CAL) on Data Bases

In order to assist the trainees on features of data bases in the DIALOG and ORBIT systems, a CAL module was designed to teach the trainees similarities and differences that may exist among the data bases as well as among systems searching the same data bases.

The module points out the importance of knowing what data elements are searchable on a given data base and emphasizes the fact that because a given data element is searchable on one data base does not guarantee that it will be searchable or even exist in another data base.

Another aspect of the search system and the data base characteristics is knowing not only what data elements exist but which are searchable using field limitations; for example, the prefix or suffix specifications of DIALOG, or the parenthesized field labels of ORBIT, which limit the search to a single data element within the data base. It is also important to understand the default search method when no data element is specified, since there are significant and different situations in different systems. The DIALOG search, when no data element restriction is specified, will proceed through to search the "basic" index of all "suffix fields", whereas the ORBIT default is to a search of all directly searchable fields, for example. Another feature of data base searching which is sensitive to search system differences is the results obtained when using multi-word terms. On DIALOG, for instance, the choice of a multi-word term without the proximity indicator (w) can produce radically different results than using the proximity indicator (w) depending on the circumstances and the indexing practice of the data base.

The value of "expanding" possible search terms is also detailed in the CAL module. Both the DIALOG data base features as well as the ORBIT features are covered in the CAL module.

Section 5 S.T.I. Training Modules Report

December 1977

Analysis of Use and Evaluation of Training Modules
Individuals Primarily Responsible: Ronal Nicholas, Elaine Caruso

and

Analysis and Display of Automatically Collected Use Data
Individual Primarily Responsible: Joan Scollon

Related Appendices:

H. Sample of User Attribute Data Collection

I. Partial Listing of Data Captured by Programs

J. Summary of User History (Preliminary Format)

K. Summary of CALP History (Preliminary Format)

L. Program Listings: Data Reduction and Display

Analysis of Use, and Evaluation, of Training Modules

We have derived a restatement of our plans for evaluation of the training program. It has two phases, relating to:

1. Workability and usability of the training programs in the end-user environment
2. Validity of the training program

First we need to be sure that the programs can be used in the environment of the end user; and that the end user will use them, that they are accepted by him and are seen to be of value. During the developmental phase just completed, we have determined that both of these tests seem to have been passed by the training program, but only in the environment of the professional searcher. To the extent that this reflects the end-user environment, we can begin to anticipate similar results in the end-user environment.

The extent of similarity is problematic; introduction of the program into end-user environments will be a focus of the next year's work. To date we can only say that the programs can be, and are, used at remote, isolated terminals, by individual users at their own initiative, without direction or interpretation by a human teacher! The motivation of a professional in information service provision is an undeniable factor in this assessment.

Our second area, the validity of the program in terms of its goals, while difficult, will be easier to accomplish. We have established our goal as the development on the part of the end user of a "first level" of proficiency:*

> First level: The end user can achieve useful and satisfying results, searching with ease and confidence on one (or two) of the more comprehensive existing document retrieval systems, in those data base files of closest interest to his subject of study and research.

---

*For a more detailed statement of this "level of proficiency" see Training Manual, p. , and see Appendix for a defined three level sequence of proficiency development

User attributes, as indicated on the data collection form, Appendix H, and demographic data related to his professional area and status, statistics on his use of the programs (see automatic data collection and analysis, next section); and his use of the operational services themselves, subsequent to the training program, in terms of the availability of the services, will be factors in the assessment.

Pending the introduction of the training program into the end-user environment, we are looking to the results obtained by information professionals, as predictors of performance, in mechanical aspects of the "proficiency" test (see Appendix I), and developing the data reduction, analysis, and display capabilities of the program.

Automatic Data Collection and Analysis

The "user snapshot" technique, which collects a set of user program and time coordinates at key stages of the trainee's progress through the computer modules, gives us a continuing, chronologically ordered log of program use. Data collected includes

1. job number, assigned by the University Computer System;
2. the identification number (computer login number) used by the trainee;
3. month, day, year of login;
4. time of day of entry into training program;
5. name of module entered or exited (I-IX);
6. name of module segment entered or exited (i.e., DIALOG, ORBIT, or in some instances, SCORPIO).

From this raw data, we plan to look for patterns of trainee progress, and patterns of module utilization. Presently available data of use from June 1, 1977 to November 30, 1977 is distorted by the inclusion of data generated by our own debugging runs and by the fact that the programs were in varying stages of completion during that time period. The data is sufficiently valid for the development and testing of data reduction and display programs, however.

User patterns will be examined on a per session basis, and on the basis of the totality of a trainee's experience with the training program. We are looking for a division of user types, according to our hypothesized categories of user learning styles, i.e., users are expected to be of the holistic or serialistic style. If we can identify such a dichotomy in our trainee population, with attendant gains in learning efficiency when they can follow their "natural" style of learning, we can take steps to insure that this natural tendency is further facilitated by the organization of the computer and printed textual materials.

Present organization of the modules is basically serial, that is learning of the search process is presented sequentially, step by incremental step. Individuals are, however, encouraged to use the modules in any sequence which their experience or inclination suggests to them; they may enter Modules VIII or IX (DIALOG and ORBIT emulations) as a first experience, if they choose; or go directly from Module III, which gives an overview of the searching process by means of a simple, guided "sample" search, to Modules VIII or IX. Either of these latter patterns would be considered to be indicative of a holistic learning style. Preliminary "eyeballing" of the data shows serialist learning styles predominant to date. No conclusions are being drawn pending better collection of user data, as noted earlier.

Module usage patterns will be analyzed to determine how much time a trainee spends on a particular module, and how often he uses it; whether it is reentered following an attempt to use the emulated system. This data will be used to improve or adjust module content to fit trainee needs.

Background to the Data Reduction and Analysis Programs

This computer assisted learning program consists,at this point, of nine modules which teach various aspects of the data bases, and the DIALOG and ORBIT interactions. The user has a manual, "Hands on On-Line in a Multi-System Multi-Data Base Trainer," to be used in addition to the online interaction. Some users were given an introductory lecture, and they completed personal data sheets at that time. From the personal data sheets, it was found that 50% were GSLIS students, 25% were faculty, and 25% were from professions outside the University of Pittsburgh; virtually all were enrolled in a course at GSLIS.

This program is still in the experimental stage and must be evaluated before it is released for general use. Users of the program have been asked to return freely annotated printouts of terminal sessions to Dr. Caruso, the project leader. Although not everyone returns printouts, enough printouts are received to give an indication of problems in the program, such as ambiguity in the text or programming errors. The project office also receives feedback from professional searchers at the Knowledge Availability Systems Center concerning how accurately CALP portrays the original systems. The immediate problem is to evaluate available, automatically recorded CALP module usage data. Problems for subsequent study are the development of methods of assessing the effectiveness of teaching methods and assessing user efficiency and effectiveness in the use of the data bases.

Specifically, CALP usage data was studied to answer the following questions. How much time are users spending in the modules? What are the patterns of CALP module usage? Does this tell us anything about the success of our program? We are interested in testing the hypothesis that there are two different approaches to module usage: serial or holistic. If this hypothesis is proven, then we must design the system with both groups in mind. We also thought that unusually short times or unusually long times spent in particular modules might indicate problems in the modules. Our purpose here was not to assess our teaching effectiveness; we wanted an overview of how people use the system.

The raw data from which the analyses were made consisted of 20 pages of data generated by the automatic data collection program (a sample of the data is included in Appendix I). Each record contains job number, project, programmer number, name (if it is the first record of a session), date, the time logged into or out of a module, and the name of the module used, in a variable length record format. The records were organized only by job number and project, programmer number, resulting in jumbled data that was almost impossible to work with from a human or machine point of view. To determine how much time users spent in each module and the patterns of module usage per session, proved to be a task which required that the data be reorganized.

The data elements were put in fixed fields so they could be sorted using the University of Pittsburgh's SORT package (examples in appendix). The data was sorted on 4 keys to arrange it chronologically by terminal session. Four FORTRAN programs were written to do the data processing.

The first program (TIMER.F4) calculated elapsed time per module by subtracting the time a module is entered from the time it is exited. The time calculation itself is a relatively simple process. But the program became complicated due to the number of conditions that had to be checked: was the module aborted?; what is the module subprogram (DIALOG, ORBIT, SCORPIO), if any?; is it a new user?; if end of session, what is total time?

TOTAL.F4 reads the times file ('TIMES.DAT') and calculates average time per session and per module. PATTRN.F4 reads the times file to determine patterns of module usage and the number of times each pattern is used. RDPAT.F4 read the random access pattern data file ('PATRN.DAT'), which was written by PATTRN.F4 and writes the data on the lineprinter or terminal.

Data Analysis and Results

Patterns of Module Usage

There are 15 different modules (including submodules) of CALP. Because the CALP index module is almost always the first entry, there are in theory approximately 14! combinations of modules used. Beyond the CALP index module, there are no requirements for order of module selection. In fact only 83 combinations occurred, as chosen by approximately 175 users (as in "terminal users"; there are not 175 unique users; some are repeats).

Only 2 users stayed at the terminal to use as many as 9 modules (they did not use the same modules in the same order). Typical users went through no more than 3 or 4 modules per session. The number of possible combinations of module usage was further reduced because SCORPIO (as a submodule) is not fully implemented across the system, and the ORBIT emulator is not available for general use.

Sessions with only 2 modules are more likely to occur than those with more because the probability of 2 independent events occurrence is greater than that of > 2 events occurring, and in this case, the appearance of the CALP index is not a random or independent event. Thus we have figures like the following:

#users	patterns of usage
35	CALP, Mod 5
28	CALP, Mod 3 DIALOG
20	CALP, Mod 1

More interesting are examples of slightly longer patterns of usage which illustrate the hypothesis that there are 2 types of program users: those who exhibit a holistic approach and those who exhibit a serial approach.

#users	patterns of usage
6	CALP, MOD 4, Mod 5
3	CALP, Mod 6 DIALOG, Mod 7
3	CALP, Mod 2, Mod 3 DIALOG
3	CALP, Mod 1, Mod 2
5	CALP, Mod 3 DIALOG, Mod 4
3	CALP, Mod 1, Mod 2, Mod 3 DIALOG, Mod 4, Mod 5
9	CALP, Mod 7, Link DIALOG emulator
2	CALP, Mod 1, Mod 2, Mod 3 DIALOG, Mod 4
8	CALP, Mod 3 DIALOG, Mod 4, Mod 5
4	CALP, Mod 1, Mod 2, Mod 3 DIALOG
5	CALP, Mod 3 DIALOG, Link DIALOG emulator
9	CALP, Mod 6 DIALOG, Mod 6 ORBIT
9	CALP, Mod 6 DIALOG, Mod 6 ORBIT

It is hard to separate users as being strictly serialistic or holistic because Modules 3 and 6 always indicate a submodule. But from this sample, it appears that 32 are serial users and 18 are holistic. From a general perusal of the pattern data it is evident that both types of users exist in great enough numbers that we should keep both groups in mind. This is an extremely tentative preliminary conclusion, but sufficiently interesting to warrant further examination of the notion.

Analysis of Summary of CALP Usage per Session

An explanation should be made first for all the zeroes in the data (see Appendix K). SCORPIO has only been implemented in Module 3 so far, and was not used. The ORBIT emulator was not then available for general use. Whenever a module is aborted, the time is recorded as 0.00 minutes. ORBIT segments of Modules 3 and 6 were aborting, being incomplete during the data collection period. The DIALOG emulator was not logging an exit time and is thus treated as an abort.

Since an abort condition may result for 3 reasons (a program execution error, the lack of an exit time recorded, or user aborting on purpose), users may be spending several minutes in these modules before the module ends. Thus average session time may be longer than 16 minutes.

It is difficult to arrive at any conclusions based on this data. Since we didn't have an idea of just how much time should be spent in each module, it is hard to make a statement about these averages. On a subjective basis, it seems that these times are somewhat short,considering the length and content of the modules. More study and analysis with the now complete modules and data collection routines is indicated.

Discussion

Many CALP users learned of the system by word of mouth and accessed the system using their own project, programmer numbers. While we do not specifically object to this, we would like to have the data we would have gotten from personal data forms which were completed at the introductory sessions. In the CALP index module, the user could be asked if he is a new user. If the answer is no, he proceeds to the next part of the module. If the answer is yes, he is asked to answer a few questions. He is given the option of refusing. Here we can inquire concerning his educational and professional background and his reasons for wanting to learn the use of the bibliographic information retrieval services. A menu selection of choices could be offered concerning these questions, so that they can be automatically coded for future processing. Then when we implement user effectiveness and efficiency measures, we can investigate the possibility of correlation between educational/professional background, motivation and other factors, and search efficiency/effectiveness.

Section 6 S.T.I. Training Modules Report

December 1977

Dissemination of the Training Program
Individual Primarily Responsible: Elaine Caruso

Related Appendices:

M. The On-Line Training Institute: Abstract of Proposal to Buhl Foundation, and Contact Brochure

N. Description of Proposed Cooperative Project, with Professor A. Neelameghan, Director, Documentation Training and Research Center, Bangalore, India

O. List of Published Articles, and Papers, Generated by/about the S.T.I. Training Modules Project

Dissemination of the Training Program

The dissemination effort has two aspects: first, the public awareness of the program is cultivated; then the program itself, or access to it, is distributed.

The effort to alert the potential users of the results and discoveries of this project has included, to date, several papers and articles (see Appendix O); most effective, to date, in all probability, was the paper delivered at the November 14-16, 1977 Pittsburgh Conference: On-Line Revolution in Libraries, attended by 700+ individuals; and the concurrent announcement of the On-Line Training Institute established at the University of Pittsburgh, which will use products of the project (see Appendix M); and distribution of 500 reprints of the article which appeared in ONLINE, OCTOBER 1977, "A Trainer for Online Bibliographic Retrieval Systems."

Use of the programs in an operational sense is perhaps most substantively and firmly exemplified in the On-Line Training Institute (above); however, the use projected, beginning in January 1978 by the Kansas City Libraries Metropolitan Information Network, T. Philip Tompkins, Project Director, is even more promising, in that the projected use is more akin to the central goal of our training program development. At the K.C.L.M.I. Network, the programs will be used by individuals at their own initiative, in their own patterns of time and rate of progress. More useful data for program evaluation is anticipated from this arrangement.

The agreement with the Network Project Director, Mr. T. Tompkins, calls for use of the programs to proceed in stages:

1. Beginning January 16, 1978, a team of three experienced searchers from the MCLMI Network will access the programs on the University of Pittsburgh Computer System, as an exploratory effort; they will

prepare a report on the programs projected usefulness in developing and maintaining a staff of searchers for the network.

2. A representative will spend a week at the University of Pittsburgh, subsequent to the exploratory effort and report, to familiarize himself with the programs and data base files, and to meet the project staff and participate in evaluation plans for the next phase of cooperation.
3. Beginning in March 1978, an account and user status will be established for the network, with the program to be used in staff training offered through the eight search site libraries to information service types.

An arrangement for implementation of the training modules at the Documentation Research and Training Center, Bangalore, India, is in the early stages of proposal development (see Appendix N).

Dissemination to end user populations is projected to a more restricted geographic area; that is within the University itself, to populations of users in the Libraries of the School of Engineering of Western Psychiatric Institute, of the Biological Sciences, and of the Health Professions; of Public and International Affairs, and to populations of students in those schools, in classes focussing on the literature of the various fields of study, and on research methodology and resources.

Appendices

Appendices A, B, D, F, G, L are separate documents available on application to the S.T.I. Project, D. E. Caruso, P.I., 732 L.I.S. Building, University of Pittsburgh, Pittsburgh, PA 15260. Pages from these documents sufficient to illustrate comments and references within the body of this report are included in this section.

Appendix A

Selected pages from
"Hands On Online in a Multi-System, Multi-Data Base Trainer--A Manual"
to be Used in Support of CALP Modules

What the Hands on Online Multisystem Multi-Data Base Trainer Is:

This is an instructional package, consisting of self-explanatory modules of computer assisted learning and practice (CALP Modules), and emulators of the DIALOG and ORBIT search systems. The emulators are training versions of the DIALOG and ORBIT systems, which allow practice of the "most-used"* capabilities of the systems. Files from various data base suppliers can be searched with the emulators, allowing trainees to develop some competence in the use of these files, as well as the fundamental facility for use of the search language.

The purpose of the program package is to bring the complete novice to a first level of competence, as defined below, and to provide a quick refresher course for the trained but infrequent user of the services.

This manual is intended for use with the CALP modules; it describes and supplements each module, and in the instance of Module III, duplicates much of the descriptive text from the module. Suggestions for practice on the emulated systems (Appendix A)are correlated with the CALP modules, to illustrate the capabilities taught, and take into account the actual content of the files available in the emulator data base. These "practice" sessions could be used with the operational online services, if accessible to the trainee, rather than on the emulated services.

---

*identified by the services themselves and our own experience

The First Level of Competence in Use of On-Line Bibliographic Retrieval Systems*

The first level is described in its three aspects:

1. The trainee knows and can use the "most-used" capabilities of the service(s) of his choice.

ON DIALOG
BEGIN_ _ _(number)
.FILE_ _ _(number)
?FILES
EXPAND_____(term)
SELECT_____(term(s))
COMBINE (NOT,AND,OR)
TYPE }
PRINT } with formats
END
DISPLAY SETS
root search (TERM?)
LOGOFF

ON ORBIT
"FILE_ _ _(name)
SS 1/C? AND,NOT,OR
"FIND
"NBR
"PRINT (ONLINE and OFFLINE with formats)
"STOP
root search (TERM:)
"HISTORY

2. The trainee can use the vocabulary negotiation capabilities of the service(s) to retrieve document references of value to him; he can decrease or increase the size of the retrieved document set in a meaningful way by using the logical operators AND, OR, NOT.

3. The trainee knows the data base(s) most useful to him:

   He can use documentation as provided by the service(s) to determine: (a) which data elements of the records can be directly searched; and (b) whether controlled or uncontrolled indexing is used in the fields he searches. He knows and can use any necessary tools for choosing search terms before getting on-line. He can bring it online for searching.

---

*Second and third levels of competence, and more complete descriptions of commands implemented and omitted in the emulators, are included as appendices to the training manual

SDC - Lockheed

DATA BASES

- ABI/INFORM
- ABIPS
- AGRICOLA
- AHL
- AIM/ARM
- APILIT/APITAT
- ART BIB.
- ASI
- BIOSIS
- CACON
- CIS
- COMPENDEX
- CDI
- EIS
- ENVIROBIB

DATA BASES

- EPA REPORTS
- ERIC
- FOUNDATION GRANTS INDEX
- GEOREF
- G.R.A.
- HISTORY ABSTRACTS
- INSPEC
- ISMEC
- LISA
- LLBA
- MARKET ABSTRACTS
- MGMT CONTENTS
- METADEX
- NTIS
- OCEANIC ABSTRACTS

DATA BASES

- PAIS
- PATELL (Psy. Abs.)
- PETROLEUM ABSTRACTS
- PHARMACEUTICAL NEWS INDEX
- POLLUTION ABSTRACTS
- PATS/PREDICASTS
- SCI SEARCH
- SOCIOLOGICAL ABSTRACTS
- SSCI

List of Available DIALOG and/or ORBIT
File Titles for Use with Emulators

December '77

EMULATORS ONLINE COMMANDS*

(Abbreviated and symbolic forms, as well as full forms,
of commands and operators are implemented)

FUNCTIONS	DIALOG	ORBIT
(1) HOUSEKEEPING		
Starting search after login	BEGIN ! BEGIN number	Directly connected to default file
Terminating search and disconnecting from system	END LOGOFF	"STOP
Asking for list of accessible data base files	BEGIN ?FILES	"FILES?
Elapsed time/cost	(Search time given automatically at end of search, BEGIN or .FILEn)	(Given automatically at login, logout, and at end of search-)
Deleting search statements no longer needed	(BEGIN or BEGINn erases set history)	(Not implemented)
(2) SEARCHING		
Entering search terms	SELECT S # , with term E numbers, range of E numbers	Usually in search mode) "FIND ___ (for bypassing program queries)
Displaying alphabetical list of terms	EXPAND E " (no thesaurus access)	"NBR
Creating search logic	COMBINE C use set numbers with AND,OR,NOT (*,+,-) operators	Usually in search mode, use AND,OR, and NOT operators)
Changing data bases	BEGIN ___ (insert file no.) . .FILE ___ (insert file number)	"FILE ___ (insert file name)
Root searching	TERM? (any number of characters)	TERM: (any number of characters)
Restricting searches	By author AU= and title /TI fields	By author (AU) and title (TI) fields

*after Lawrence,Prewitt,Bearman, "User's Quick Guide", NFAIS, Philadelphia, March 1977 (see List of Aids, Appendix 3)

EMULATORS ON-LINE COMMANDS (p.2)

FUNCTIONS	DIALOG	ORBIT
(3) Output		
On-line printing	TYPE T ' (for hard copy terminals)use format options	"PRINT "PRT "P "PRINT TRIAL "PRINT FULL
Formatting printout	Up to 10 defined formats for each file--see file descriptions. Specify set number,format,items)	"PRINT ___,___,___, ___ (specify set number, format instruction, field acronym, items)
Off-line printing	PRINT PR & (with format options)	"PRINT OFF-LINE (used as "PRINT above)
Interrupting on-line output	Available, but different from DIALOG (CONTROL-O)	Available but different from ORBIT (CONTROL-O)
Erasing whole lines	Available, but different (CONTROL U)	$ and Carriage Return
Erasing single character	Delete key	Delete key
(4) SUPPORT FEATURES		
Explanation	EXPLAIN - refers trainee to CALP modules	"EXPLAIN - refers trainee to CALP modules
Providing description of search	DISPLAY SETS DS @	"HISTORY(displays list of search statements, with all ORBIT options, e.g., range: "HISTORY 6-10
Assistance on procedures	Local phone, WATS line	Local phone, WATS line

COMMANDS NOT IMPLEMENTED

FUNCTIONS	DIALOG	ORBIT
(1) HOUSEKEEPING		
Determining elapsed time/cost	.COST - but see table of implemented commands	"TIME "TIME INTERVAL "TIME RESET - but see table of implemented commands
(2) SEARCHING		
Deleting search statements no longer needed	DIALOG does not use; 99 statements allowed (however BEGINn erases all sets)	"ERASEALL "BACKUP __ "KEEP ___ "RESTART
Root searching	Partially implemented; use of repeated ?'s not yet implemented	TERM# (for single character)(but see table of implemented commands)
Text searching	(Suffix Fields Only) FULLFILE word proximity search. Most used indicators: TERM (w) TERM (adjacent) TERM (nw) TERM (n intervening terms) TERM (F) TERM (in same field) TERM (c) TERM (in same citation)	STRINGSEARCH (searches for words or imbedded character strings) SENSEARCH (searches for words or imbedded character strings in a sentence) (in specified fields of previously created sets only)
Stacking commands	SCHILDREN;SAGGRESSION; STELEVISION; C1-3/ AND; T4/6/1-3 (does not presently accept multiple commands on one line)	
Saving search strategies	END/SAVE .RECALL ___ (serial number assigned at SAVE) .EXECUTE ___ (serial number) .RELEASE (to delete a RECALL'ed SAVE'd search statement) END/SDI for automatic run of search at end update	"KEEP(to save previous) "SAVE(to save subsequent) "KEEP(SS #'s) "SAVE CANCEL "DELETE (search name) "PURGE ________
Saving search results		
Saving search results	KEEP __/__ (set no/item nos.)	NA

COMMANDS NOT IMPLEMENTED (p.2)

FUNCTIONS	DIALOG	ORBIT
(2) SEARCHING (Continued)		
Restricting searches	LIMIT LIMITALL (dates, language, accession numbers, special features) RANGE SEARCHING	Date ranging
Creating search logic	Set-range short cut: C1-4/AND Stacking commands: Sterm; Sterm	"SUBS "SUBHEADINGS (stores and applies to all terms untill cancell
(3) OUTPUT		
Sorting	.SORT (only selected data bases and fields)	"PRINT OFF-LINE SORT ___ (category names)
Interrupting on-line output	Break Key (but see table on implemented commands)	Break Key (but see table of implemented commands)
Erasing single characters	Backward slash - varies (but see table of implemented commands)	Control # - varies (but see table of implemented commands)
(4) SUPPORT FEATURES		
Requesting system news providing explanation of data bases, commands, etc. Assistance on procedure	?NEWS EXPLAIN ___, ? ___ Toll free phone number (but see table of implemented commands)	"NEWS "EXPLAIN (last message) "EXPLAIN ___ (name) "HELP (but see table of implemented commands) "COMMENT "DIAGRAM "DIAG (SS#) "ORDER "RENAME ___ TO ___ "SECURITY (to get password "SECURITY CANCEL "STORAD (for mailing off-line printouts) "VERSION { short, "VERS } symbolic

Appendix B

Selected pages from

"Suggestions for Independent Practice"

to be Used in Extension of CALP Experiences

Exercises for Independent Practice

Most users of the training modules (CALP 1-7) like to use what they learn in the modules as quickly and "realistically" as possible. "Real" use in the online service is prohibitively expensive, however; so we have provided emulators--imitations of the search service for initial practice sessions.

Suggestions for making these self-guided efforts more efficient are given in the exercises provided for Modules, 1, 3-7. The exercises reflect contents of the modules and of the sample data base used by the emulators of the DIALOG and ORBIT systems (Modules 8 and 9).

Before using emulators, or any retrieval system:

1. Always come with a "question" appropriate to the available files; study file description; write the questions out.
2. If you have access to vocabulary aids for the files, use them.
3. List all possible search terms for each search aspect, or concept, both from aids and other sources, including your own knowledge of the topic.

In all the following suggested exercises we will suggest search content, but you should also improvise questions to extend your understanding as you gain control of the more mechanical aspects of the searching process.

Exercises for Use Following Module I

Object: CALPI taught you to follow very rigidly and precisely a particular sequence of steps. Inevitably you will make errors. Learn how to deal with them! and begin to discover where and how you can vary your approach to the computer to suit yourself.

1. Type a misspelling of login. Note computer response. Type a misspelling of login, but before you hit the carriage return, hold down the CONTROL key (on your left) and strike the "U" key. This erases the line before it is sent to the computer, and thus you get no error message. Remember this use of CONTROL-U.

2. Variations and shortcuts in use of system commands are often available to "initiates" of the computer. Try this variation of the commands you learned in Module I.

   ```
   .LOG _ _ _ _ _ _ _/_ _ _ _ _ _
   .RUN CTLYST
   ```

   How does it differ from the protocol you were taught?

3. Deliberately enter an erroneous password. Now try to recover from the error to complete your login. If you become frustrated, you can hold the CONTROL key down while striking C to start over from the beginning; the value of this effort lies in your persisting until you succeed.

4. When CATALYST asks for the lesson and section number, use (i.e., MISUSE) letters o and l for the zero-zero-one. Is the message the computer gives you helpful in understanding your "error"?

When the computer (either the operating system or a special program such as the CALP modules) is waiting for you to enter a command or respond to a question, it has a list of likely responses which is compared with whatever you enter. If the response or command you enter does not match any of the pre-stored expected entries, you will be told so using previously stored messages. If the program is very "sophisticated" the message can be very specific about why your entry failed to produce the results you expected it to; otherwise you get a very generalized error message. In some cases the computer program cannot detect your error at all. This can result in erroneous or misleading answers in which you may have no way of recognizing incorrect or incomplete responses. We will point out such cases in later modules. A very simple example:

> Failure to spell a search term correctly will simply cause the search programs to report 0 results; or if you use one of several spelling variations (film,films) you might retrieve about half of the appropriate documents (where film=93 items, films=87 items).

Suggestions for Exploratory and Self Testing Use of the Emulators

Each of the CALP modules III-VII introduces and gives practice in use of search service commands and capabilities; you should test yourself for mastery by trying what you have learned on the emulated search system(s), and by freely exploring the effects of each new technique on the search process.

However, use of the emulated systems, or the systems themselves, requires that you think not just of the command language skills and techniques, but also, in a quite different way, of the subject content of the file, and of questions to be put to the file. This is analogous to putting a beginning driver whose feet and hands don't yet know where clutch, brake, gas feed, and turn signals are, out on a strange highway.

The exercises are suggestions, as a roadmap, to help you more readily find your way; the emulators are the training grounds where penalties for errors are not so costly as on the cross country turnpikes.

If you have access to DIALOG or ORBIT on a training-use arrangement you can, of course, do your practicing in the "real" system. We suggest in any event that you practice first on the "emulated" system, and then use the "real" system.

MODULE III Suggestions for Emulator Practice

Objective: to be able to use the "most used" commands (see Manual), from memory, in running searches on the system(s) of your choice.

In actual practice you will analyze questions, identify concepts and try to match the query to a particular data base file (or files). You will use any aids you have available, to determine what are the best search terms, and the logical connectors to best combine them.

For now, to allow you to first concentrate on the use of the system, we will give you some searches which will be productive in the small files associated with the emulators.

Choose Module VIII (DIALOG) or Module IX (ORBIT) and call up one or more of the data base files, running the searches suggested below. You might want to examine documentation describing each file before you begin. (AND,OR,NOT, when underscored, are logical connectors, not part of the search term.)

NTIS
1. POINT
2. SOLAR ENERGY
3. SOLAR AND ENERGY
4. SOLAR OR NUCLEAR

PAPER CHEMISTRY
1. PAPER
2. PAPER MAKING
3. PAPER AND MAKING
4. PAPER OR PAPER MAKING

SOCIOLOGY ABSTRACTS
1. SOCIAL CHANGE AND HEALTH
2. SOCIAL CHANGE OR HEALTH
3. SOCIAL PROBLEMS AND SOCIAL CONDITIONS IN RUSSIA

If you have access to DIALOG or ORBIT for training purposes, try the same searches in the larger files which are available there. (Note: Paper Chemistry is up on ORBIT only.) If you see any differences in your output which are not attributable to the fact that more and different citations are in the file, find out why the differences occur. (Study file descriptions, not changes in indexing procedures; look at portions of the index by using "NBR or EXPAND commands. If you have access to an experienced searcher, discuss your experience with him.)

MODULE IV Data Bases: Suggested Practice Searches

Objective:

1. to be able to "call up" different files in the data base
   e.g., DIALOG: FILE3 BEGIN2
   e.g., ORBIT: "FILE NTIS
2. to be able to use field restrictions when entering search terms
   e.g., DIALOG: AU=JONES, J? COLORIMETRY/TI
   e.g., ORBIT: JONES, J: (AU) COLORIMETRY(TI)

1. Read the descriptions provided (See Manual) of the data base files provided with the CALP emulators. Read also any other available documentation of those files.

   Note that emulators are only capable of handling author and title categories, and the unrestricted search of the DIALOG Basic Index and the ORBIT comprehensive index. (For further explanation of the Basic Index and the comprehensive indexes, see documentation provided by the respective systems.) In Module V we will look at implications for search term choices, resulting from variations in system policies for creating the different indexes.

2. Run these searches, using field restrictions as appropriate, on either or both emulators.

   PAPER CHEMISTRY
   1. All articles on PAPER MAKING or MANUFACTURING
   2. Only articles where PAPER MAKING or MANUFACTURING is the indexer's choice as subject entry for the article
   3. Articles on FIBERS
   4. An article by Bernard Ivanovich Aikhodzhaev
   5. An article by I. L. Lami

   NTIS
   1. All items on AIR TRAFFIC and AIR TRAFFIC CONTROL
   2. All items on LITHIUM COMPOUNDS
   3. Only items where LITHIUM COMPOUNDS is the indexed subject of the document
   4. An article by Kei Priestley

3. If you have access to DIALOG or ORBIT for training purposes, try restricting your search to other directly searchable fields. Go back to the file descriptions for aid in creating your searches before you go online.

   Compare search results when the same subject search terms are used, but (1) restricted to title field, or (2) to index term field, or (3) to abstract field, or (4) used without restriction.

MODULE V Negotiating Search Terms

Objective: Becoming skilled at use of browsing capabilities of the system(s) to choose all necessary forms of terms for creating an effective, efficient search. More specifically stated, the objectives of these exercises are:

1. to be able to review alphabetically near terms,
   e.g., DIALOG
   ```
   EXPAND AU=JOHNSTON, T    E ANTIPHLOGISTEN
   ```
   e.g., ORBIT
   ```
   "NBR JOHNSTON, T(AU)     "NBR PHLOGISTINE
   ```

2. to be able to use displays in specifying search terms,
   e.g., DIALOG
   ```
   S E6-8    (choosing 3 displayed name forms relating to
              wanted author)
   ```
   e.g., ORBIT
   ```
   "FIND JOHNSTON,THAD:    (using most specific truncated form of
                            name)
   ```
3. to be able to enter multiword search terms for retrieval from either multiword indexed fields or single word indexed fields,
   e.g., DIALOG
   ```
   S FAMILY AND MARITAL COUNSELING     (multiword field)
   S FAMILY (W) MARITAL (W) COUNSELING  (single or multiword fields)
   S FAMILY: S MARITAL: S COUNSELING  } single or multi-
   C 1 AND 2 AND 3                    } word fields
   ```
   e.g., ORBIT
   ```
   FAMILY AND MARITAL COUNSELING          (multiword fields)
   FAMILY AND MARITAL AND COUNSELING      (single word fields)
   ```

Use of all available vocabulary aids is always good practice, prior to going online to search; however many of the useful directly searchable fields are not "indexed" by any controlled technique and thus the printed tools cannot help you to discover useful terms for stating your question. Author names, words from the title, notes and abstract fields are put into the index just as they happen to be on the original tapes, so vocabulary aids and subject lists are of little help with these fields. The use of truncated forms of author names, and to capture variant forms of a search term is very useful and efficient for these fields where no vocabulary control is exercised; however, it can result in unwanted documents, if a truncated form is not wisely chosen. Use of "NBR or EXPAND commands, while it will cost you extra online time, may be worthwhile, if it helps you to identify more precise forms, eliminating excessive numbers of unwanted items from your output.

The file segments used in the emulations of DIALOG and ORBIT are too brief to give you a realistic experience of the uses of browsing and truncating in a full-scale data base. We suggest you limit your practice of these capabilities to an exploration, by use of the browsing commands of the actual inclusion which would occur if you use truncated terms to create sets. For example, create sets by using truncated forms of varying lengths of the search terms used in the previous exercise, then use the EXPAND or "NBR command to see the actual terms used in the sets you created with the truncated form.

1. On either system, try the truncated form to create a set, then use EXPAND or "NBR to see what terms you "hit" on.

   PAPER CHEMISTRY   PAP? (or PAP:)   CELL?
   PRO:
   Sociological Abstracts   SO?   SOC?   SOCIO?   SOCIA?

2. To discover how indexing by multiword terms can cause difficulty in retrieval, try to find all documents having to do with FIBERS in the PAPER CHEMISTRY file. There are 8 index terms, in addition to FIBER and FIBERS, which will lead you to documents on fibers; one such is SEED FIBERS.

3. Equally critical to your search may be your awareness that some fields included in the searchable index are created "automatically", by simply entering the <u>separate</u> words as they occur on the tape file whereas other fields will be included in the indexer as single <u>or</u> multiword terms as assigned by the indexer of the document.

   To see how this can affect your search results, choose a concept which would "normally" be expressed as a multiword term. Compare results of searches combining (OR) all of the separate words of the term with use of the multiword term alone; try using the single words combined with AND.

   DIALOG and ORBIT both have "full-text" search capabilities, more expensive to use than the directly searchable index, that allow you to retrieve all the instances of occurence of the multiword term, <u>wherever</u> it is in the unit record.

```
DIALOG      S      HOT(W)DOG
ORBIT       SS/1   HOT
            SS/2   SENS ,' (TI) : DOG
```

   Text search capabilities are not available on the emulator. The best you can do to get the most efficient search for a multiword term, in a field which only offers single word searchable terms, is to use the logical operator AND to combine the separate words. (In fact, ORBIT recommends this approach, rather than the most expensive and more time consuming SENTENCE limited SENS command.)

   Use some of the searches suggested below to test the effects of these file building decisions by the search services, and the use of OR and AND logic in "getting around" the difficulty.

   PAPER CHEMISTRY
   1. PAPER
   2. MAKING
   3. PAPER MAKING
   4. PAPER <u>AND</u> MAKING <u>OR</u> PAPER MAKING

4. If you have access to DIALOG or ORBIT, on a reduced rate for training basis, try these exercises (1-3, above); also devise similar exercises, by looking at the indexes ("NBR or EXPAND) to find interesting multiword and singleword terms.

MODULE VI Use of Logical Operators

Objective:

1. to be able to use AND, NOT, OR to increase or reduce size of retrieved document set meaningfully.

2. to be able to control the order in which the logical operators are executed in complex strategy statements.

1. One useful way of thinking about the use of the OR operator to connect or combine search terms, is to consider that you are creating a broader (or more general) index term than the file itself provides, and one that is more tailored to your particular interests. Your first step is thus to be sure the broad term you want does not already exist in the file. Use the description of the PAPER CHEMISTRY file provided.

   FIBER
   FIBER OR FIBERS
   FIBER OR FIBERS OR BAST FIBERS OR CELLULOSE FIBERS

2. An equally useful way of thinking about the use of the AND operator is to consider that you are creating a new index entry term which is more specific than any which is used by the data base file you are searching.

   Refer to the file description for SOCIOLOGICAL ABSTRACTS, then try the following searches and others like them which you can devise.

   (Search terms and suggested strategies are tailored for different trainee populations and inserted here.)

3. One way of thinking about the NOT operator is to consider that you are creating a narrower, more specialized form of an already existing search or index term. Try these and similar searches; note that you are reducing your output, as you did in 2 above, but this time on the basis of specific document attributes not wanted, rather than the opposite, that specific aspects are required.

(Search terms and suggested strategies are tailored for different trainee populations and inserted here.)

4. The order in which the search system executes the logical operators AND NOT and OR can make a vast difference in the meaning of your search statement. It is useful to have in your memory a model search statement which you can compare to any complex strategy (one which combines two or more different logical operators) to be sure you get the results you intend. One which concerns concrete subjects, easily visualized, is best:

   Suppose that you are searching a recipe file for a dessert which uses apples or cherries in a pastry or cake; you don't want a coffee cake or Danish pastry since these don't suit your notion of "dessert."

   To be sure you get what you want, you could select each term separately as a first step. (In DIALOG-type search systems you have no alternative; you must first select any term(s) you want to combine.) So you would have 6 sets of retrieved documents:

   1. apples
   2. cherries
   3. cakes
   4. pastries
   5. Danish
   6. breakfast

   In the next step (or as a first step in ORBIT-type systems) you could say:

   APPLES OR CHERRIES AND CAKE OR PASTRY NOT DANISH OR BREAKFAST

   which seems a straightforward statement of what you are looking for.

   But let's see what happens when the search system processes that statement. Following its rules for precedence in executing the logical operators, DIALOG would first process all pairs connected by NOT, then all pairs connected by AND, then all pairs connected by OR.

   This means we would get recipes for all PASTRIES which are not DANISH, first, and for CHERRY CAKES and all recipes using APPLES, and any recipes for BREAKFAST dishes--when what we wanted was only pastries or cakes which used apples or cherries, and were not Danish or breakfast types.

   In other words, for this strategy we wanted the OR's executed first, then the NOT applied to the PASTRY or CAKE set (we didn't want to eliminate DANISH APPLES or BREAKFAST CHERRIES!) and then the fruit set AND'ed to the cleaned up PASTRY-CAKE set.

In DIALOG-type searches which allow you to use parentheses, we could say this:

(A OR C) AND ((P OR CAKE) NOT (D OR B))

and pairs within parentheses would be combined first.

In ORBIT-type searching we are obliged to create the complex strategy in stages:

A OR C	creating set AC
P OR CAKE	creating set PC
D OR B	creating set DB
PC NOT DB	creating set PC-DB
PC-DB and AC	

To recapitulate and to make the point that precedence simply needs to be known and used appropriately (i.e., it is not wrong, just not always right), consider a search for recipe ideas where you want pastries or cheese and fruit, but are not interested in making French pastries.

PASTRIES NOT FRENCH or CHEESE AND FRUIT

is just fine; but if you want pastries, but not cheese or French or Danish pastries, then

PASTRIES NOT FRENCH OR CHEESE OR DANISH

would not be right; you'd get all the cheese pastry, and the Danish pastry, and many other cheese and Danish recipes!

(See note following Item 2)

5. To belabour the precedence further, we should note that ORBIT and DIALOG do not use the same order! ORBIT processes AND's, then NOT'ed pairs, then OR'ed pairs. DIALOG processes NOT's, then AND's, then OR's. Does this difference make a difference?

(See note following Item 2)

MODULE VI Suggestions for Emulator Practice

Remember the order in which the logical operators are executed:

in DIALOG	NOT'ed pairs, then AND'ed pairs, then OR'ed pairs (remember N A T O)
in ORBIT	AND'ed, then NOT'ed, then OR'ed (remember A N O, Ano Domini!)

1. (DIALOG or ORBIT) Test the effects of OR and AND operations in increasing set sizes (broadening your search) and decreasing set sizes (narrowing or refining your search).

   Use "synonyms" from the table at the end of this exercise, using terms from columns A and B to practice OR operation (or think up your own synonymous terms or related concepts).

   Use limiting or constraining aspects from column C to AND concepts (or devise your own search terms, using simple concepts which combined with other concepts create (1) a new concept, or (2) a particular perspective or treatment of the first concept.

2. (DIALOG) Select a range of E numbers from an EXPAND display. What logical operation was used in making the resulting SET? Create an equivalent set by using SELECT with a truncated form of your search term.

or (ORBIT) Create a set using the results of an "NBR" command. Make an equivalent set by using a truncated form of the search term. Make the same equivalent set by using the keyword ALL with the truncated search term.

   Repeat this exercise until you get a good sense of the value of the two approaches, i.e., examination of the indexes followed by specific choices of search terms, or blanket choices based on roots (or stems) of search terms.

3. (ORBIT) Test the effect of the order of execution of the AND, NOT, OR operations by running searches like these:

   SSI A OR B AND C *
   SS2 A OR B
   SS3 2 AND C

   Have SS1 and 2 printed out for comparison. Be sure you understand why they differ.

---

*Choose terms from the table or "make up" appropriate "questions".

or (DIALOG) Test the effect of the order of execution of the NOT, AND, OR operations by running searches like these:

SELECT A *
SELECT B
SELECT C
Combine A OR B AND C
COMBINE (A OR B) AND C

Have sets 1 and 2 printed out for comparison. Be sure you understand why they differ.

---

*Choose terms from the table or "make up" appropriate "questions"

MODULE VII   Suggestions for Emulator Practice

Objective:

1. To be able to get search results printed online or offline, specifying which sets, which and how many items, in what formats they should be printed.

2. To be able to specify appropriate formats for output, when one simply wants to sample results, to identify documents for hard copy acquisition, to get terms for broadening a search.

1. (DIALOG) Using results of any search you care to make, practice creating variations of the TYPE format: changing the three arguments SET/FORMAT/ITEMS. Note what happens when you fail to specify one or more of the arguments. This is called the "default" option. Default options should produce most commonly useful results.
Try a TYPE command specifying only the set number. What is the format and item default? (Remember San Francisco Interchange to remember the order of the options--SET/FORMAT/ITEMS.)
Try format 1, 2 without specifying items. How many items are given you?
Try repeating this step. What items do you get?

or (ORBIT) Using any search result, try "PRINT FULL and "PRINT TRIAL option. What differences do you observe?
Try the default option, by just using "PRINT (or "PRT) without specifying either FULL or TRIAL. What are the differences in fields displayed? In number of items?
When no particular SS (Search Statement) is specified (e.g., "PRT) the last set created is printed. Practice printing other sets: "PRT SS1.
Try specifying which fields are to be printed:

```
"PRT  TI, AU  SS2
```

2. (DIALOG or ORBIT) Run a search, and have the results printed out in each format, specifying the same document to be printed each time. Repeat the exercise, using another file. Note differences between formats, and if there are differences in the same format when another file is used.

3. (DIALOG) TYPE commands give you online output. The PRINT or PR command, is used to request offline printout of search results. It is used with the same format arguments as the TYPE command. Try it, as you used TYPE, to get a "feel" for it online. No offline prints will actually be generated in the emulator/DIALOG system.

or (ORBIT) ORBIT uses the same command for online or offline printout; simply add the specification OFFLINE to any form of the PRINT statement (sorting can only be done for offline printing).

Appendix C

Illustration of Use of CALP Modules by Trainees

TYMNET TERMINAL MODEL IDENTIFIERS

IDENTIFYING CHARACTER	TERMINAL DEVICE
A	DATAPOINT 3300, INFOTON, AND DIGILOG, HAZELTINE, AND ALL OTHER CRT TERMINALS
B	MODEL 37 TELETYPE
C	UNIVAC DCT-500

CONTINUE THE ACCESSING PROCESS FROM THIS POINT.

```
 PLEASE TYPE YOUR TERMINAL IDENTIFIER
>A
-1072-03--
PLEASE LOGIN:>LRS
USER NAME:
>DIALOG
HOST IS ON LINE


     GOOD, YOU CAN TRY IT AGAIN IF YOU WANT MORE PRACTICE.
IF SO, TYPE M. IF NOT, TYPE N.

>>M
 PLEASE TYPE YOUR TERMINAL IDENTIFIER
>C
   YOU SHOULD TYPE  A  .
 PLEASE TYPE YOUR TERMINAL IDENTIFIER
>A
-1072-03--
PLEASE LOGIN:>LRS
USER NAME:
>DIALOG
HOST IS ON LINE
```

```
     OKAY. IN THIS EXAMPLE YOU ARE ACCESSING DIALOG VIA
THE TELENET NETWORK. YOU ARE USING A TEKTRONIX 4023
TERMINAL, AND YOU ARE AT THE POINT AFTER PLACING THE
TELEPHONE HANDSET INTO THE ACOUSTIC COUPLER. DUE TO
CERTAIN OPERATING CHARACTERISTICS OF THE LANGUAGE
BEING USED TO WRITE THIS TRAINING PACKAGE, WE CANNOT
SIMULATE THAT PART OF THE  TELENET PROCEDURE
CALLING FOR 2 RETURNS AT THE START.
THEREFORE ASSUME THAT YOU HAVE ALREADY ENTERED
THE 2 RETURNS.

     BELOW IS A PARTIAL LISTING OF THE TELENET
TERMINAL IDENTIFIERS.


            TELENET TERMINAL MODEL IDENTIFIERS
            ==================================


UNBUFFERED TERMINAL MODELS                  IDENTIFIER
--------------------------                  ----------


    DATA 100 73                             DT73
    DATAPOINT 3300                          DP3300
    HAZELTINE 2000                          HZ2000
    TEKTRONIX 4023                          TK23
    UNIVAC DCT-500                          UV500




TELENET
415 DL9

TERMINAL=>TK23

@>DIALOG

    C 415 20 IS THE DIALOG COMPUTER NUMBER.

@>C 415 20


415 20   CONNECTED


    GOOD. IF YOU WANT TO TRY THIS ACCESSING ROUTINE AGAIN
TO GAIN MORE FAMILIARITY WITH THE PROCEDURES, TYPE  A,
TO GO ON,  TYPE   G.
>>A
```

108

```
     FOR YOUR SEARCH ON THE TERM "AIRCRAFT SIMULATION",
YOU WOULD PROBABLY SELECT DATABASE NUMBER 6, THE NTIS
DATABASE. WITH THIS IN MIND, CONTINUE.

?>6
     YOU MUST NOW TYPE BEGIN6 TO BEGIN YOUR SEARCH AND
TO SELECT THE NTIS DATABASE.
?>BEGIN6

     NOW YOU WOULD PROBABLY WANT TO EXPAND YOUR PROPOSED
SEARCH TERM. TRY IT BELOW:

            15 JAN 77   15:13:04   USER 478
     $0.57  0.009  HOURS IN FILE6
FILE 06*NTIS           1964-1977      ISS01
               SET   ITEMS    DESCRIPTION
               ===   =====    ===========
?>HELP
   YOU MUST ENTER  EXPAND  OR  E  AIRCRAFT SIMULATION
TO EXPAND THE TERM.
               SET   ITEMS    DESCRIPTION
               ===   =====    ===========
?>EAIRCRAFT SIMULATIOPN
   YOU MUST ENTER  EXPAND  OR  E  AIRCRAFT SIMULATION
TO EXPAND THE TERM.
               SET   ITEMS    DESCRIPTION
               ===   =====    ===========
?>E AIRCRAFT SIMULATION
REF     INDEX-TERM                          TYPE  ITEMS  RT
 E1     AIRCRAFT-----------------------------     2074
 E2     AIRCRAFT DESIGN----------------------      123
 E3     AIRCRAFT FLIGHT TESTING--------------       72
 E4     AIRCRAFT MAINTENANCE-----------------      127
 E5     AIRCRAFT RESEARCH--------------------      273
 E6    -AIRCRAFT SIMULATION------------------       97
 E7     AIRCRAFT TESTING---------------------      147
 .

E16     AIRFOIL------------------------------      247


     NOW LET'S ASSUME THAT YOU WANT TO  SELECT  AIR-
CRAFT RESEARCH   AS TERM 1, AND   AIRCRAFT SIMULATION
AS TERM 2.

?>HELP

     YOU SHOULD ENTER  S E5  OR  S AIRCRAFT RESEARCH
TO SELECT THE FIRST TERM.
?>SE5
              1      273    E5:AIRCRAFT RESEARCH
?>SE6
              2       97    E6:AIRCRAFT SIMULATION
```

Now let's try a runthrough of a simple search. We will assume that you have already logged into the ORBIT system, and that you want to begin searching on the term SPACE STATION.

```
SS 1/C?
USER:
>*NBR SPACE STATION
     You should type *FILES?
>*FILES?

PROG:
     YOU MAY ACCESS THE SSIE, CHEMCON, ERIC, NTIS, DEMO NTIS,
AND POLLUTION DATABASES.
YOU ARE NOW CONNECTED TO THE DEMO NTIS DATABASE.
```

Recall that you should use the FILES? command first when using the ORBIT emulator on this system.

Since our search term will be SPACE STATION, the NTIS database would appear to be the most reasonable choice. Select it below.

```
SS 1/C?
USER:
>*FILE NTIS

PROG:
YOU ARE NOW CONNECTED TO THE NTIS DATABASE.
```

Now that we are connected to the NTIS database, let's use the NBR command with our term SPACE STATION to find its precise stored form as a Descriptor(DE).

```
SS 1/C?
USER:
*NBR> SPACE STATION
     You should type *NBR SPACE STATION (DE).
>*NBR SPACE STATION(DE)

PROG:
POSTINGS            TERM
   422            SPACE SHUTTLE
  1147            SPACE SATTELITES
     1            SPACE STATION
   232            SPACE STATIONS
    11            SPACE SURVEYS
UP N OR DOWN N?--HOW MANY TERMS UP OR DOWN?
USER:
```

By using the NBR command we found that the term SPACE STATION was used only once as a Descriptor (this was probably a misspelling). The term that we really want to use is the plural SPACE STATIONS. Now that we know this correct form, try typing it in as a search term.

STANDARD SEARCH TERM ENTRY

ORBIT normally does not use any special command for entering search terms. The normal procedure is to type in the term after the normal system prompt. Say, for example, we want to enter the term NUCLEAR RESEARCH(IT).

```
SS 1/C?
USER:
NUCLEAR RESEARCH(IT)

PROG:
SS     1     PSTG(723)
```

We typed in our search term, NUCLEAR RESEARCH(IT), after the program prompted us. Then, the search was run and the program notified us that 723 items were found to be indexed by our term (PSTG 723).

Try typing in the term USSR(WL)-WL means Work Location.

```
SS 2/C?
USER:
>USSR(WL)

PROG:
SS     2     PSTG(8475)
```

The program tells you that there are 8475 items listing USSR in the Work Location category.

FIND command

The FIND command is used to enter a search term when the program does not present the normal prompt. We have already seen such a situation with the UP N OR DOWN N message. Using the FIND command as follows allows you to enter a search term instead of answering the program's question.

```
UP N OR DOWN N?--HOW MANY TERMS UP OR DOWN?
USER:
·FIND MATURATION(IT)

PROG:
SS     3     PSTG(324)
```

FILES? command
======

This command instructs the program to type a list of the names of all the databases that are available to you at the time you are searching. It is entered as follows:

```
SS 1/C?
USER:
*FILES?
```

Try typing in the FILES? command below.

```
SS 1/C?
USER:
*FILES?
>
PROG:

YOU MAY ACCESS THE SSIE, CHEMCON, ERIC, NTIS, DEMO NTIS,
AND POLLUTION DATABASES.
YOU ARE NOW CONNECTED TO THE DEMO NTIS DATABASE.
```

What you see above is the sort of listing that the program will type in response to the FILES? command. You can then pick the database that you want to search.

Many time you will already know the database that you want to search, along with the times when it is available. When that is the case, you won't need to use the FILES? command. On the ORBIT emulator in this training package, however, you should always use the FILES? command right after you get on the emulator to see what databases are available.

FILE command
====

This command allows you to choose the database you want to search. It can be used at the beginning of a search session to choose the first database to be searched or, it can be used during a session to switch from one database to another. If used to switch databases, however, all results from searching the old database will be wiped out when the switch is made to the new database. Therefore, all printing of search results from the old database must be done before the switch.

The command is entered:

```
SS 1/C?
USER:
*FILE ERIC
```

Suppose you want to see how the term CRYSTAL is stored as an Index Term(IT). You would enter the NBR command, then CRYSTAL(IT). Try it below.

```
SS 8/C?
USER:
>'NBR CRYSTAL(IT)

PROG:
POSTINGS          TERM
     1         CRYSTAAL
     1         CRYSTABOLITE
 18888         CRYSTAL
    11         CRYSTAL/CS
     1         CRYSTALIZATION
UP N OR DOWN N?--HOW MANY TERMS UP OR DOWN?
USER:
```

Now try telling the program to type the next 5 items by typing DOWN 5 below.

```
UP N OR DOWN N?--HOW MANY TERMS UP OR DOWN?
USER:
>'DOWN 5
     TYpe DOWN 5 to see the next 5 terms.

UP N OR DOWN N?--HOW MANY TERMS UP OR DOWN?
USER:
>DOWN 5

PROG:
POSTINGS             TERM
    11            CRYSTALFIELD
     1            CRYSTALFREE
     1            CRYSTALIN
     1            CRYSTALINE
     2            CRYSTALINITY
UP N OR DOWN N?--HOW MANY TERMS UP OR DOWN?
USER:
```

Above you see an example of the display of the next 5 alphabetical terms. Note that misspellings are also caught by using the NBR command (i.e., CRYSTAAL,CRYSTALIN).

Appendix E

Excerpts from Runs of DIALOG™ and ORBIT™ Emulators*

*DIALOG™ is the trademark for the online bibliographic retrieval service offered by Lockheed Retrieval Services, Palo Alto, CA, and ORBIT™ is the name of the service offered by System Development Corp., Santa Monica, CA

```
ENTER YOUR DIALOG PASSWORD>
         LOGON FILE 1 13:57:25


?>/
                                  INVALID COMMAND

?>///
                                  INVALID COMMAND

?>?
PLEASE REFER TO THE CAL MODULES FOR COMMAND EXPLANATIONS
TO EXIT FROM *DIALOG* TYPE    LOGOFF
?>BEGIN 1
        18-Aug-77 13:58:26.5USER
 $0.00      .002 HOURS IN FILE 1
FILE 1:PAPER CHEMISTRY

           SET ITEMS DESCRIPTION
           --- ----- -----------

?>BEGIN 2
        18-Aug-77 13:58:38.0USER
 $0.00      .037 HOURS IN FILE 1
FILE 2:NTIS

           SET ITEMS DESCRIPTION
           --- ----- -----------

?>S AIR
            1    2   AIR

?>EX ACCIDENT?
 REF   INDEX-TERM              TYPE ITEMS RT
E 1   WITTIG CURT            -------     1
E 2   WOLF JACK KEIL         -------     1
E 3   WONG A Y               -------     1
E 4   WORDSLANGUAGE          -------     1
E 5   WRITE                  -------     1
E 6  -X ACCIDENT?            -------
E 7   X RAY DIFFRACTION      -------     4
E 8   XII                    -------     1
E 9   XRAY                   -------     2
E10   XRAYS                  -------     1
E11   Z                      -------     1
E12   ZADEH L A              -------     1
E13   ZEROGAP                -------     1
E14   ZIMMERMAN GEORGE L     -------     1
E15   ZINC                   -------     1
E16   ZIRCONIUM              -------     1


?>E ACCIDENT?
 REF   INDEX-TERM              TYPE ITEMS RT
E 1   ABSORBER               -------     1
E 2   ABSORPTION             -------     .
```

```
HELLO FROM PIT/ORBIT. (27-Jun-77  16:48 46.4  LOCAL TIME)
YOU ARE NOW CONNECTED TO THE PAPER CHEMISTRY     DATABASE.

ARE YOU A NEW USER?  IF YES ENTER Y.  IF NO ENTER N OR A COMMAND.


USER:
>'NBR TERM

PROG:
POSTINGS    TERM
      1     TEAR
      4     TEMPERATURE
      1     TENSILE
      1     TEST
      1     TESTING
TERM                    DOES NOT EXIST IN THE INDEX IN THIS FORM

UP N OR DOWN N?
USER:
>UP 4

PROG:
POSTINGS    TERM
      1     SYMPOSIA
      5     SYNTHESIS
      2     SYSTEMS
      1     TABLES
      1     TEAR
UP N OR DOWN N?
USER:
>'NBR COMPUTER

PROG:
POSTINGS    TERM
      8     COMPOUNDS
      1     COMPUTATION
      1     COMPUTER
      2     COMPUTERS
      4     CONCENTRATION
UP N OR DOWN N?
USER:
>'STOP

PROG:
ALL DONE? (YES/NO)
USER:
>NO

PROG:
SS 1 /C?
USER:
```

```
?>S ACCIDENT?
               2    3   ACCIDENT

?>S ACCIDENTS
               3    3   ACCIDENTS

?>S AIRCRAFT?
               4    0   AIRCRAFT

?>C 1 AND (2 OR 3)
               5    3    1 AND (2 OR 3)
?>T
INVALID SET VALUE
?>T1

AD-A015 172/0SL
 Enroute Air Traffic Control Program Software Optimization Study.
 Prince, L. D.
 Logicon Inc San Pedro Calif*Federal Aviation Administration, Washingt
on, D.C. Systems Research and Development Service.
 *Air traffic control systems, *Management planning and control, *Comp
uter programming  Optimization, Data storage systems, Central processi
ng units, Coding, Mathematical analysis, Utilization, Computer program
s, Data processing,
 FAA-RD-75-87
 Jun 75,  229p
 DOT-FA74WA-3457
?>TY1/5/1-3
INVALID SET VALUE
?>T1/6/1-3

AD-A015 172/0SL
 Enroute Air Traffic Control Program Software Optimization Study.

AD-A015 138/1SL
 Operational Test and Evaluation of U.S. Army Forward Area Air Defense
s.
?>T1/5/1-3

AD-A015 172/0SL
 Enroute Air Traffic Control Program Software Optimization Study.
 Prince, L. D.
 Logicon Inc San Pedro Calif*Federal Aviation Administration, Washingt
on, D.C. Systems Research and Development Service.
 *Air traffic control systems, *Management planning and control, *Comp
uter programming  Optimization, Data storage systems, Central processi
ng units, Coding, Mathematical analysis, Utilization, Computer program
s, Data processing,
 FAA-RD-75-87
 Jun 75,  229p
 DOT-FA74WA-3457
 This report documents results of a 10-month study aimed at reduc
he core storage and CPU utilization required by the NAS Enrout
erational software. Re               esented for six major 
nitor improvements                             e improvements
men   mathema                                     izati
```

```
?>S ATOM?
            18    3   ATOM

?>C 17AND18
            19    3    17AND18
?>C19NOT1
            20    3    19NOT1
?>S TEMPER?
            21    4   TEMPER

?>C20NOT21
ARGUMENT SYNTAX ERROR
?>C 20NOT21
SET HAS NO MEMBERS
?>C 20AND21
            22    1    20AND21
?>T
INVALID SET VALUE
?>T21

AD-A015 171/2SL
 Compatibility Guide for Adjacent Loading of Bulk Liquid Cargoes.
 National Academy of Sciences Washington D C Committee on Hazardous Ma
terials*Coast Guard, Washington, D.C. Office of Research and Developme
nt.
 *Compatibility, *Chemicals, *Marine transportation  Hazards, Liquids,
 Safety, Accidents, Tables(Data), Cargo, Temperature, Gases, Pressure,
 Water pollution, Abatement, Handling, Storage, Co
 USCG-D-156-75
 Feb 75,  33p
 DOT-CG-41680-A
?>PRINT 22
PRINT
COMMAND NOT CURRENTLY SUPPORTED
?>S POLLUT?
            23    0   POLLUT

?>S GAS?
            24    4   GAS

?>T24/5/1-4

AD-A015 123/3SL
 Strongly-Cooled Ionizing Plasma Flows with Application to Venus,
 Ong, R. S. B.
 Michigan Univ Ann Arbor Dept of Aerospace Engineering*Air Force Offic
e of Scientific Research, Arlington, Va.*Oxfo      v. (England). Mathe
mat
                  res, *Solar wind  P                     ing, Venus
                      eractions.                                 lect
                             ing
```

118

```
ENTER YOUR DIALOG PASSWORD>JGW00123
         LOGON FILE 1 08:27:14

?>.FILES
PLEASE ENTER THE FOLLOWING INFORMATION.
PRESS 'RETURN' FOLLOWING EACH ENTRY.
-----------------------------------------
SEARCH TITLE>TEST OF NTIS DATABASE
SEARCHER NAME>JAMES G. WILLIAMS
REQUESTER NAME>ELAINE CARUSOE
MAIL ADDRESS                             N/A

YOU MAY ACCESS THE FOLLOWING FILES:
      1--PAPER CHEMISTRY
      2--NTIS DATABASE
      3--

ENTER NUMBER OF DESIRED FILE>2


TITLE                    TEST OF NTIS DATABAS
DATE/FILE               12-May-77 / 2
SEARCHER                 JAMES G. WILLIAMS
REQUESTER                ELAINE CARUSOE
ADDRESS                  N/A
            SET ITEMS DESCRIPTION
            --- ----- -----------

?>E WAVES
 REF    INDEX-TERM                 TYPE ITEMS RT
E 1    TIME                      --------    1
E 2    TRAINING                  --------    1
E 3    TRANSFORMATION            --------    1
E 4    TRANSITIONS               --------    1
E 5    VISUALIZATION             --------    1
E 6   -WAVES                     --------    1
E 7    WIND                      --------    1

?>E ALUMINA
 REF    INDEX-TERM                 TYPE ITEMS RT
E 1    AFTERGLOWS                --------    1
E 2    AIRPORTS                  --------    1
E 3    ALCOHOLS                  --------    1
E 4    ALIGNMENT                 --------    1
E 5    ALKANES                   --------    1
E 6   -ALUMINA                   --------    1
E 7    ALUMINUM                  --------    2
E 8    ANTENNA                   --------    1
E 9    ARSENIC                   --------    1
E10    ATOMS                     --------    1
E11    ATTENUATORS               --------    1
E12    AUGER                     --------    1
E13    AUTOMATION                --------    1
```

```
?>SE7
             7    2   E7

?>SELECT CHEN\N\MICAL
             8    3   CHEMICAL

?>COMBINE 7 AND 8
SET HAS NO MEMBERS
?>COMBINE 7 OR 8
             9    5    7 OR 8
?>DS
            SET ITEMS DESCRIPTION
            --- ----- -----------

             1    0    CHLORINATION
             2    0    E6
             3    2    E3
             4    2    E11
             5    6    E9
             6    2    5 AND 3
             7    2    E7
             8    3    CHEMICAL
             9    5    7 OR 8
?>.FILE 1
         13-May-77 13:04:06.0USER
 $0.00    0.000 HOURS IN FILE 2
FILE 2:PAPER CHEMISTRY

            SET ITEMS DESCRIPTION
            --- ----------------

?>EXXPAND CHLORIDE
 REF   INDEX-TERM              TYPE ITEMS RT
E 1   WAVE                   --------     1
E 2   WHEAT                  --------     1
E 3   WHITE                  --------     1
E 4   WOOD                   --------     1
E 5   WORMS                  --------     1
E 6  -XXPAND CHLORIDE        --------
E 7   XYLANS                 --------     1
E 8   XYLOSE                 --------     1
E 9   YEASTS                 --------     1

?>.FILE2
         13-May-77 13:04:48.8USER
 $0.00    0.000 HOURS IN FILE 1
FILE 1:NTIS DATABASE

            SET ITEMS DESCRIPTION
            --- ----------------
```

```
?>T 12/2

AD-A015 097/9SL
 Mechanisms and Inhibition of Dealloying in an Alpha Brass.
 Fort, Wil
 Florida Univ Gainesville Dept of Materials Science and Engineering*Of
fice of Naval Research, Arlington, Va.
 18
 18
 Aug 75,  211p
 N00014-68-A-0173-0015
?>DS
             SET ITEMS DESCRIPTION
             --- ----- -----------

              1    0    CHLORINATION
              2    0    E6
              3    2    E3
              4    2    E11
              5    6    E9
              6    2    5 AND 3
              7    2    E7
              8    3    CHEMICAL
              9    5    7 OR 8
             10    3    SOLU
             11    1    TETRA
             12    1    ZINC
?>COMBINE11 AND 12
SET HAS NO MEMBERS
?>COMBINE 10 AND 12
             13    1    10 AND 12
?>T13/5

AD-A015 097/9SL
 Mechanisms and Inhibition of Dealloying in an Alpha Brass.
 Fort, Wil
 Florida Univ Gainesville Dept of Materials Science and Engineering*Of
fice of Naval Research, Arlington, Va.
 18
 18
 Aug 75,  211p
 N00014-68-A-0173-0015
 Dealloying of a single-phase binary alpha brass in pH 4, 0.1M chlorid
e solution was investigated in terms of the operative mechanism. Selec
tive leaching of zinc from the brass was produced under conditions of
cathodic polarization. Auger electron spectroscopy was employed to mea
sure composition versus depth profiles of the resulting zinc-depleted
surface layers. The selective leaching mechanism was found to be based
 on enhanced volume diffusion of zinc to the brass surface. Dealloying
 by dissolution of the brass as a whole, followed by redepos
?>.FILE2
          13-May-77 13:11:58.1USER
 $0.00    0.000 HOURS IN FILE 2
FILE 2:NTIS DATABASE
```

```
HELLO FROM PITT/ORBIT. (14-Dec-77  11:59 40.7  LOCAL TIME)
YOU ARE NOW CONNECTED TO THE PAPER CHEMISTRY     DATABASE.

ARE YOU A NEW USER?  IF YES ENTER Y.  IF NO ENTER N OR A COMMAND.


USER:
>'FIND BOOKS

PROG:
SS  1 PSTGS ( 4)
SS  2/C?
USER:
>PRINTING

PROG:
SS  2 PSTGS ( 1)
SS  3/C?
USER:
>'PRINT SS 1

PROG:

AN-  46-12
TI-  POLLUTION ECOLOGY OF FRESHWATER INVERTEBRATES,
AU-  Hart, C. W., Jr.; Fuller, S. L. H.
SO-  Acad. Press (N.Y. & London), 389 p. (1974).

AN-  46-84
TI-  FLAME RETARDANCY OF POLYMERIC MATERIALS,
AU-  Kuryla, W. C.; Papa, A. J.
SO-  Marcel Dekker Inc. (N.Y. 10016), 2 vol., 317 + 242 p. (1975). ($5
     7.50)

AN-  46-101
TI-  INTRODUCTION TO OPTIMIZATION METHODS,
AU-  Adby, P. R.; Dempster, M. A. H.
SO-  Wiley & Sons (Halsted Press, N.Y.); Chapman & Hall (London), 204
     p. (c1974).

AN-  46-13
TI-  PHYTOCHROME.  (Proceedings of a symposium held at Eretria, Greece
     , Sept. 1971),
AU-  Mitrakos, K.; Shropshire, W., Jr.
SO-  Acad. Press (N.Y. & London), 631 p. (c1972).
SS  3/C?
USER:
>'PRINT

PROG:

AN-  46-107
TI-  EMPLOYMENT OF MATHEMATICAL-STATISTICAL METHODS IN TECHNOLOGICAL P
     ROCESSES,
AU-  Riedl, R.
SO-  Papier Druck (Allg. Teil) 23, no. 3:  41-43; no. 4:  53-56 (March
USER:
>•FIND BOOKS

PROG:
SS  1 PSTGS ( 4)
SS  2/C?
USER:
>•FILES?

PROG:
YOU MAY ACCESS THE PAPER CHEMISTRY
NTIS
AND SOC ABSTRACTS       DATABASES
YOU ARE NOW CONNECTED TO THE PAPER CHEMISTRY     DATABASE
SS  2/C?
USER:
>•FIND PRINTING

PROG:
SS  2 PSTGS ( 1)
SS  3/C?
USER:
>PRINTS

PROG:
SS  3 PSTGS ( 2)
SS  4/C?
USER:
>•HISTORY

PROG:


SS  1:  BOOKS                                                      ( 4)
SS  2:  PRINTING                                                   ( 1)
SS  3:  PRINTS                                                     ( 2)


SS  4/C?
USER:
>•PRINT TRIAL

PROG:

TI-  MEASUREMENT OF OPTICAL UNEVENNESS,
DE-  CONTRAST,  ELECTROMAGNETIC RADIATION,  LIGHT,  MEASURING INSTRUME
     NTS, MOTTLE,  OPTICAL INSTRUMENTS,  OPTICAL PROPERTIES,  PERCEPTI
     ON,  PRINTING PAPERS,  PRINTS,  RADIATION,  REFLECTANCE,  REFLECT
     OMETERS,  SCANNING, SENSORS,  SMOOTHNESS

TI-  WEYERHAEUSER (CO.)'S NEW PHOTO CENTER HAS CENTRAL OFFICE, MICROFI
     LM SYSTEM,
DE-  CATALOGS,  COLLECTION,  DATA PROCESSING,  DATA RECORDING,  DATA S
     TORAGE, DOCUMENTS,  FILM,  INFORMATION RETRIEVAL,  MATERIALS HAND
     LING,  MICROFILM, PHOTOGRAPHIC SLIDES,  PHOTOGRAPHS,  PHOTOGRAPHY
     ,  PRINTS,  RECORDING, STORAGE,  SUPPLY,  UNITED STATES,  WASHING
     TON,
SS  3/C?
USER:
>*PRINT SS 1 TRIAL

PROG:

TI-  POLLUTION ECOLOGY OF FRESHWATER INVERTEBRATES.
DE-  ANIMALS, BIOLOGY, BOOKS, CRUSTACEANS, DOCUMENTS, ECOLOGY, E
     FFLUENTS, FRESH WATER, INSECTS, INVERTEBRATES, MICROORGANISMS,
       POLLUTION, PROTOZOA, WATER, WATER ANIMALS, WATER POLLUTION,
      WORMS.

TI-  FLAME RETARDANCY OF POLYMERIC MATERIALS.
DE-  ACETATES, ADDITION POLYMERS, AMINE POLYMERS, ANIMAL FIBERS, B
     OOKS, CHLORIDES, COMPOSITES, COPOLYMERS, DOCUMENTS, ELASTOMER
     S, FIRE PROOF MATERIALS, FLAME RETARDANTS, FORMULATIONS, HALI
     DES, HALOGEN COMPOUNDS, INORGANIC COMPOUNDS, NATURAL FIBERS, N
     ITROGEN COMPOUNDS, NYLON, ORGANIC COMPOUNDS, PHOSPHORUS COMPOU
     NDS, POLYAMIDES, POLYCONDENSATES, POLYETHYLENE, POLYHALOCARBON
     S, POLYHYDROCARBONS, PO
[illegible]
USER:
*PRINT TRIAL

PROG:

TI-  EMPLOYMENT OF MATHEMATICAL-STATISTICAL METHODS IN TECHNOLOGICAL P
     ROCESSES.
DE-  DISTRIBUTION, MATHEMATICAL ANALYSIS, MATHEMATICS, PARAMETERS,
      PREDICTION, PRINTING, PROBABILITY, PROCESS CONTR
[illegible]
USER:
*PRINT SS 1 FULL 2

PROG:

AN   46-12
TI-  POLLUTION ECOLOGY OF FRESHWATER INVERTEBRATES,
AU-  Hart, C. W., Jr.; Fuller, S. L. H.
SO-  Acad. Press (N.Y. & London), 389 p. (1974).
DE-  ANIMALS, BIOLOGY, BOOKS, CRUSTACEANS, DOCUMENTS, ECOLOGY, E
     FFLUENTS, FRESH WATER, INSECTS, INVERTEBRATES, MICROORGANISMS,
       POLLUTION, PROTOZOA, WATER, WATER ANIMALS, WATER POLLUTION,
      WORMS.
AB-  The aquatic biology and ecology of protozoans, sponges, flatworms
     , leeches, earthworms, bryozoans, crayfishes, clams, mussels, sna
     ils and water insects are dealt with in 10 chapters by different
      experts with special emphasis on the effects of effluents and p
     ollution on these organisms. A subject index is appended.

AN-  46-84
TI-  FLAME RETARDANCY OF POLYMERIC MATERIALS,
AU-  Kuryla, W. C.; Papa, A. J.
SO-  Marcel Dekker Inc. (N.Y. 10016), 2 vol., 317 + 242 p. (1975). ($5
     7.50)
DE-  ACETATES, ADDITION POLYMERS, AMINE POLYMERS, ANIMAL FIBERS,
     B  KS, CHLORIDES, COMPOSIT     COPOLYMERS, DOCUMENTS, EL
        FIRE PROOF MATERIAL           RETARDANTS, FORMULAT
          HALOGEN COMPOUN                        MPOUNDS, NATUR
        GEN COMPOUNDS,                           UNDS, P
          POLYAMIDES
          LYHYDRO
           has
```

```
SS  3/C?
USER:
>'PRINT FULL

PROG:

AN-  46-107
TI-  EMPLOYMENT OF MATHEMATICAL-STATISTICAL METHODS IN TECHNOLOGICAL P
     ROCESSES,
AU-  Riedl, R.
SO-  Papier Druck (Allg. Teil) 23, no. 3:  41-43; no. 4:  53-56 (March
     -April 1974). (Ger.)
DE-  DISTRIBUTION,  MATHEMATICAL ANALYSIS,  MATHEMATICS,  PARAMETERS,
      PRECISION, PRINTING,  PROBABILITY,  PROCESS CONTR
AB-  Mathematical-statistical methods make it possible to master the t
     echnological processes of production and to conduct them better.
      Statistical analysis of the processes furnishes the laws governi
     ng the process parameters, enabling one to recognize and eliminat
     e the causes for disturbances and to increase the stability of th
     e process.  In this article, the concepts of probability, random
     variables, and normal distribution are explained along with their
      application to technological processes, which are shown to be st
     ochastic rather than deterministic.  The technological importance
      of normal distribution is pointed out in connection with the stu
     dy of exactness and stability of processes. The considerations ar
     e supported by examples from printing.  (7 fig., 6 ref., 1 tab.)
SS  3/C?
USER:
>'PRINT SS 1 AU,TI 3

PROG:

AU-  Hart, C. W., Jr.; Fuller, S. L. H.
TI-  POLLUTION ECOLOGY OF FRESHWATER INVERTEBRATES,

AU-  Kuryla, W. C.; Papa, A. J.
TI-  FLAME RETARDANCY OF POLYMERIC MATERIALS,

AU-  Adby, P. R.; Dempster, M. A. H.
TI-  INTRODUCTION TO OPTIMIZATION METHODS,
SS  3/C?
USER:
>'PRINT AU,TI

PROG:

AU-  Riedl, R.
TI-  EMPLOYMENT OF MATHEMATICAL-STATISTICAL METHODS IN TECHNOLOGICAL P
     ROCESSES,
SS  3/C?
USER:
>'STOP

PROG:
ALL DONE? (YES/NO)
USER:
>YES

PROG:
GOOD-BYE!
```

Appendix H

Data Collection Form: Participant (i.e., Trainee) Background

and

Appendix I

Automatic Online Data Collection

Participant Background

Please complete this form so we can be better able to meet the needs of the particular groups defined by the answers to these items:

Name____Position/Title______________________

Campus/Business Address____Phone____________

1. Have you made use of search services before, either for yourself or your clients (use of a negotiated search facility such as MEDLINE or an Information Utilization Laboratory as the SSIUL)?

___No
___Yes, once
___Yes, a few times
___Yes, often

2a. Have you conducted searches before, either for yourself or your clients, using a computerized retrieval system?

___No
___Yes

b. Indicate for each system below, by checking the appropriate blanks, your experience and your desires for this training package:

System	Experience				Desire		
	Never Used	Training Sessions	Some Use	Much Use	Would like to learn	Would like "brush-up"	Don't Care
DIALOG							
ORBIT							
PIRETS							
B.R.S.							

3. If you are interested in some specific data bases and/or files or have specific items or questions you would like addressed at the training session, please indicate:____

____

____

```
44, "[134703,325621][redacted]          , 19 AUG 1977 10 10 22.8 CALP ENTRY"
44, "[134703,325621] 19 AUG 1977 10 20 48.8 MOD 1 EXIT"
44, "[134703,325621] 19 AUG 1977 10 12 44.7 MOD 1 ENTRY"
47, "[134703,325621][redacted]          , 29 AUG 1977 14 44 46.3 CALP ENTRY"
47, "[134703,325621] 29 AUG 1977 15 27 34.2 MOD 4 EXIT"
47, "[134703,325621] 29 AUG 1977 15 27 43.1 MOD 5 ENTRY"
47, "[134703,325621] 29 AUG 1977 15 34 06.9 MOD 5 EXIT"
47, "[134703,325621] 29 AUG 1977 15 35 02.9 MOD 6 DIALOG ENTRY"
47, "[134703,325621] 29 AUG 1977 15 41 24.5 MOD 6 DIALOG EXIT"
47, "[134703,325621] 29 AUG 1977 15 42 12.8 MOD 7 ENTRY"
47, "[134703,325621] 29 AUG 1977 15 50 52.3 MOD 7 EXIT"
47, "[134703,325621] 29 AUG 1977 15 50 58.9 LINK TO DIALOG EMULATOR"
47, "[134703,325621] 29 AUG 1977 15 09 04.8 MOD 2 EXIT"
47, "[134703,325621] 29 AUG 1977 15 09 43.6 MOD 3 DIALOG ENTRY"
47, "[134703,325621] 29 AUG 1977 15 17 40.2 MOD 3 DIALOG EXIT"
47, "[134703,325621] 29 AUG 1977 15 17 51.8 MOD 4 ENTRY"
47, "[134703,325621] 29 AUG 1977 15 05 17.8 MOD 1 EXIT"
47, "[134703,325621] 29 AUG 1977 15 05 26.9 MOD 2 ENTRY"
47, "[134703,325621] 29 AUG 1977 14 45 41.5 MOD 1 ENTRY"
49, "[134703,325621][redacted]          , 19 AUG 1977 14 41 48.2 CALP ENTRY"
49, "[134703,325621] 19 AUG 1977 14 58 08.2 MOD 2 ENTRY"
49, "[134703,325621] 19 AUG 1977 15 02 10.8 MOD 2 EXIT"
49, "[134703,325621] 19 AUG 1977 15 03 06.6 MOD 3 DIALOG ENTRY"
49, "[134703,325621] 19 AUG 1977 15 10 22.9 MOD 3 DIALOG EXIT"
49, "[134703,325621] 19 AUG 1977 15 10 35.2 MOD 4 ENTRY"
49, "[134703,325621] 19 AUG 1977 15 22 00.6 MOD 4 EXIT"
49, "[134703,325621] 19 AUG 1977 15 22 09.5 MOD 5 ENTRY"
49, "[134703,325621] 19 AUG 1977 15 24 12.2 MOD 5 EXIT"
49, "[134703,325621] 19 AUG 1977 14 57 22.9 MOD 3 DIALOG EXIT"
49, "[134703,325621] 19 AUG 1977 14 50 00.4 MOD 3 DIALOG ENTRY"
49, "[134703,325621] 19 AUG 1977 14 49 21.8 MOD 2 EXIT"
49, "[134703,325621] 19 AUG 1977 14 43 23.1 MOD 2 ENTRY"
```

user name

```
25, "[134704,325621][redacted]          , 25 SEP 1977 16 03 41.0 CALP ENTRY"
25, "[134704,325621] 25 SEP 1977 16 10 46.5 CALP EXIT"
25, "[134704,325621]STARK          , 25 SEP 1977 16 09 32.8 CALP ENTRY"
25, "[134704,325621] 25 SEP 1977 16 04 45.5 LINK TO DIALOG EMULATOR"
```

Appendix J

Summary of User History -- Preliminary Format

Appendix K

Summary of CALP History -- Preliminary Format

These represent early versions of data reduction program output, using 'snapshot' data collected during trainee runs of CALP modules

Appendix J

Summary of User History

Preliminary Format

```
MAIN. PROGRAM SIZE IS 1334 (OCTAL)

EXIT

.EX TIMER.F4
LOADING

TIMER 3K CORE
EXECUTION

56, "[134057,120121]KW         .  CALP    ELAPSED TIME: 2.73 MINUTES. 20 09 1977

56, "[134057,120121]              MOD 1   JOB ABORTED                 20 09 1977

***                               TOTAL   SESSION TIME: 2.73 MINUTES.

15, "[134703,130211]DUNCAN     .  CALP    ELAPSED TIME: 1.20 MINUTES. 22 09 1977

15, "[134703,130211]              MOD 1   ELAPSED TIME: 4.07 MINUTES. 22 09 1977

15, "[134703,130211]              MOD 2   ELAPSED TIME: 3.05 MINUTES. 22 09 1977

***                               TOTAL   SESSION TIME: 8.32 MINUTES.

20, "[134703,130211]DUNCAN     .  CALP    ELAPSED TIME: 1.30 MINUTES. 23 09 1977

20, "[134703,130211]              MOD 2   ELAPSED TIME: 2.65 MINUTES. 23 09 1977

20, "[134703,130211]              MOD 4   JOB ABORTED                 23 09 1977

***                               TOTAL   SESSION TIME: 3.95 MINUTES.

20, "[134703,130211]DUNCAN     .  CALP    ELAPSED TIME: 1.85 MINUTES. 23 09 1977

20, "[134703,130211]              MOD 1   ELAPSED TIME: 3.07 MINUTES. 23 09 1977

20, "[134703,130211]              MOD 2   ELAPSED TIME: 2.80 MINUTES. 23 09 1977

20, "[134703,130211]              MOD 3   ELAPSED TIME: 7.42 MINUTES. 23 09 1977

20, "[134703,130211]              MOD 4   JOB ABORTED                 23 09 1977

***                               TOTAL   SESSION TIME: 5.13 MINUTES.

20, "[134703,130211]DUNCAN     .  CALP    ELAPSED TIME: 1.08 MINUTES. 23 09 1977

20, "[134703,130211]              MOD 5   ELAPSED TIME: 5.08 MINUTES. 23 09 1977

20, "[134703,130211]              MOD 6   ELAPSED TIME: 4.78 MINUTES. 23 09 1977

20, "[134703,130211]              MOD 7   ELAPSED TIME: 7.78 MINUTES. 23 09 1977

***                               TOTAL   SESSION TIME: 8.73 MINUTES.

NUMBER OF USERS:       5
****
EXECUTION SUMMARY:

CPU TIME= 0.46  ELAPSED TIME= 1:24.38
NO EXECUTION ERRORS DETECTED

FORSE(32B)    LIB40(32.3)        KI-10
```

```
**ERROR IN [illegible]***
                              OD 4                  0.00   ?<- DATA SHIFTED - CORREC

                    ***SUMMARY OF CALP USAGE PER SESSION


AVERAGE TIME CALP USAGE:                    6.14 MINUTES.

AVERAGE TIME MOD 1 USAGE:                   6.53 MINUTES.

AVERAGE TIME MOD 2 USAGE:                   4.66 MINUTES.

AVERAGE TIME MOD 3 USAGE:                  11.61 MINUTES.
AV TIME MOD 3 DIALOG USAGE:                11.61 MINUTES.
AV TIME MOD 3 ORBIT USAGE:                  0.00 MINUTES.
AV TIME MOD 3 SCORPIO USAGE:                0.00 MINUTES.

AVERAGE TIME MOD 4 USAGE:                   6.45 MINUTES.

AVERAGE TIME MOD 5 USAGE:                   5.84 MINUTES.

AVERAGE TIME MOD 6 USAGE:                   3.93 MINUTES.
AV TIME MOD 6 DIALOG USAGE:                 5.62 MINUTES.
AV TIME MOD 6 ORBIT USAGE:                  0.84 MINUTES.
AV TIME MOD 6 SCORPIO USAGE:                0.00 MINUTES.

AVERAGE TIME MOD 7 USAGE:                   7.77 MINUTES.

AV TIME DIALOG EMULATOR:                    0.00 MINUTES.

AV TIME ORBIT EMULATOR:                     0.00 MINUTES.

AV TIME SCORPIO EMULATOR:                   0.00 MINUTES.

AVERAGE SESSION TIME:                      15.64 MINUTES.
```

Appendix K

Summary of CALP History

Preliminary Format

Appendix L

Abstracts of FORTRAN Programs

(Program Listings Available; Reference
at Beginning of Appendices)

ABSTRACTS OF FORTRAN PROGRAMS

SPACES.F4

This program reads the prime data file('STI.USE') created by the data acquisitions program. If the record is the first record of the session (i.e., if a name appears in spaces 21-35), the rest of the record, starting with the first numerical character is moved to the right to begin in column 36. Other records are left intact. It writes the file called 'NEW.DAT'. **NOTE: Records that were not the first in a session were shifted using UPDATE before this program was written. This program could be modified to align those items also.

TIMER.F4

This program reads the sorted data file ('USERS.DAT') and writes 'TIMES.DAT'. The elements of a record are read into a temporary location and compared to those of the subsequent record. Depending on various conditions, time is calculated and written as time per module. On a per session basis, module times are added together to give total time. Errors (conditions which are not anticipated) are printed on the lineprinter or terminal

TOTAL.F4

This program reads 'TIMES.DAT' and writes output on the lineprinter (batch job) or terminal (interactive). It adds times together on a per module and submodule basis (DIALOG, ORBIT, SCORPIO) and divides these times by the number of users of the appropriate module to produce average time per module and average time per session (average of the TOTAL session times).

PATTRN.F4

This program reads 'TIMES.DAT' and writes 'PATRN.DAT', a random access file. It reads the module and submodule fields of each record in a terminal session and writes a record which contains only the module and submodule sequence per session and a counter. Each record is compared against all other records previously written in PATRN.DAT. If a match occurs, the counter is incremented by 1. If no match occurs, the record is written in the next available space.

RDPAT.F4

This program reads patterns of module usage (PATRN.F4) and writes these patterns and the number of times these module usage patterns are used on the lineprinter or terminal.

MISCELLANEOUS PROCEDURES

TO CHANGE PRIME DATA FILE TO UPPER CASE:

```
.R UPDATE
INPUT==> ↲
> $UP
> $INPUT=>STI.USE
> $END
> CATALOG NAME==> STI.USE
```

TO SORT FILE BY DATE, JOB, PPN, TIME:

```
. R SORT 24K
*USERS.DAT=STI.USE/KUNA 40.2,UNA37,2,UNA2.2,UXA8.13,UXA48.8/X85
```

**Because of the size of this file, it should be run in batch mode. Specify core requirement on $JOB card (24K).

Appendix N

Description of Proposed Cooperative Project, with Professor A. Neelameghan, Director of Documentation Training and Research Center

1. Since it is apparent that on-line retrieval systems as presently developed are likely to become available before India can develop a capability to support its own systems, it is deemed appropriate to begin *first* to develop a population of users capable of exploiting the existing services.

I To that end we propose a joint effort to establish an on-line training center, which could be a part of the INSDOC operation at Bangalore, or at the Department of Science and Technology at Delhi, which could include versions of our computer programs, implemented on the facility used by Professor Neelameghan and Professor Devadason and their students at Bangalore, at a similar level to that evidenced by their work which we saw on our visit. Textual and visual materials and computer programs are presently being developed here, for such a center to train in-service librarians over a brief period of time. Our experience in this development, adapting our individualized approach to small group use, could be further extended for use in a formal, for-credit course in a degree-granting program. I believe Indian training of librarians and information specialists is closely akin to ours, in content and degree of expertise developed, so that such development would be mutually enriching.

We are planning currently for the implementation of the computer assisted instructional modules, the emulators of DIALOG and ORBIT, and the data base files on our own minicomputer; the experience gained in getting a version of the computer programs running in an outside environment would be of great value to our development; as a learning experience for our Indian counterparts it could in itself be a most cost-effective way of comprehending the technology of the data base retrieval system.

II A second effort of cooperation, to be pursued with INSDOC, if interest develops, would be the extension of the computer instruction programs, texts and visuals, for a course of study or workshop to develop an understanding of the whole range of computer applications in libraries, but especially

1. circulation control
2. cataloging and acquisition processing
3. computer storage and retrieval

with major emphasis on the generation, structuring, formatting, and updating of the data bases, per se. Standardization of input processing and data element organization is an important early lesson to be learned.

Appendix O

Published Articles and Papers on S.T.I. Training Modules

140

Attachment to Form 98a, Item 8, Summary, List of Publications,

N.S.F. Grant No. DSI 76-09538

1. Caruso, Elaine, Training Manual, "Hands On Online in a Multi-System Multi-Data Base Trainer," October 1977
2. Caruso, Elaine and Griffiths, John, "A Trainer for Online Systems," ONLINE, v. 1, no. 3(October 1977) pp. 28-34
3. Caruso, Elaine, "Training (End) Users of On-Line Services," accepted October 1977 for publication in On-Line Review
4. Caruso, Elaine, Training Modules for Users of Online Retrieval Service, Paper presented at Indo-U.S. Seminar on Scientific and Technical Information, Bangalore, India, July 18-22, 1977
5. Caruso, Elaine, "Training and Retraining of Librarians and Users," Preprint of paper presented at 1977 Pittsburgh Conference on the On-Line Revolution in Libraries, November 14-16, 1977. (To be included in conference proceedings, Marcel Dekker, publisher, 1978)
6. First Report of Progress, N.S.F. Grant No. DSI 76-09538, "Training Modules for use of Scientific and Technical Information Services," August 1976 - October 1976
7. Second Report of Progress, "Training Modules for Use of Scientific and Technical Information Services," January 1977
8. Third Report of Progress, "Training Modules for Use of Scientific and Technical Information Services," May 31, 1977

An informal presentation of the Training Modules Project was made at the ASIS Midyear Meeting:

Elaine Caruso as speaker and panelist in "ASIS Workshop: Online Reference Retrieval Workshop for Effective Use" at ASIS Midyear Meeting, Syracuse University, May 18, 1977

141